

Exhibit G

From: John Fisher <phxfish@gmail.com>
Sent: Tue, 29 Oct 2019 15:28:50 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: "Conner, Gayle" <gayle.conner@haynesboone.com>, Alfonso Íñiguez <alfonso@swarmtechnology.us>
[cisco response 10292019.pdf2.pdf](#)

I have attached a letter in response to your letter of October 11, 2019.
John Fisher



Virus-free. www.avast.com

EXHIBIT

Iniguez 19 5/14/21 DJ

exhibitsticker.com



October 29, 2019

Via US Mail and email to theo.foster@haynesboone.com

Theo Foster
 Haynes and Boone, LLP
 2505 North Plano Road, Suite 4000
 Richardson, TX 75082

Re: Swarm Technology Licensing Opportunity FRE 408

Dear Mr. Foster:

This letter is in response to your letter of October 11, 2019. In your letter you expressed your disagreement with Swarm's characterizations. Swarm continues to disagree with your objections for the reasons set forth below.

First:

You again took issue with Swarm's identification of "controller" in the Cisco device. This issue can be summarized by asking whether column A below on the left, which sets forth the '004 claim language is equivalent to column B on the right which describes the Cisco device as described in Cisco references.

A Patent '004 Claim Element	B Cisco's Device
a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;	<p>The administrator for the PnP server sets device authentication mechanisms which are acceptable for a particular deployment. [Reference 1, page 18]</p> <p>Deployment related operational tasks:</p> <ul style="list-style-type: none"> • Establishing initial network connectivity for the device • Delivering device configuration • Delivering software and firmware images

(480) 319-2233

phxfish@gmail.com

8300 S. Homestead Lane, Tempe, AZ 85284



	<ul style="list-style-type: none"> • Delivering licenses • Delivering deployment script files • Provisioning local credentials • Notifying other management systems about deployment related events [Reference 1, page 5]
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

In order to answer, we must expand on the descriptions of A and B.

A '004 patent	B Cisco Device
<p>The “controller” is also referred to as “CPU”:</p> <p>“includes a controller (CPU) 402” (’004 Patent, Column 11, lines 36)</p> <p>The controller is used to execute a software program:</p> <p>“The CPU 11 may be any single or multi-core processor, applications processor or microcontroller, used to execute a software program.” (’004 Patent, Column 5, lines 17-19)</p> <p>By virtue of running a software, the controller populates the task pool with tasks:</p> <p>“a controller configured to populate the task pool” (’004 Patent, Column 12, lines 37-38)</p>	<p>The Cisco DNA Center controller includes a PnP Server, see Figure 1 (below).</p> <p>The Cisco DNA Center controller is used to execute a software program:</p> <p>“Cisco DNA Center software resides on the Cisco DNA Center appliance and controls all of your Cisco devices” [Cisco DNA Center, page 3]</p> <p>By virtue of running a software program the Cisco DNA Center controller populates the PnP Server with tasks to auto-provision devices with images and configuration.</p> <p>In the example shown below, the Cisco DNA Center controller populates the integrated PnP Server:</p> <p>Here is an example: The PnP agent is a Cisco Catalyst switch with factory-default settings. The switch leverages the built-in day-0 mechanism to communicate with Cisco DNA Center and support the integrated PnP server function. Cisco DNA Center dynamically builds the PnP profile and configuration sets that enable complete day-0 automation.</p>



	(Cisco DNA Center SD-Access LAN Automation Deployment Guide, page 3-4)
--	------------------------------------------------------------------------

Table 1

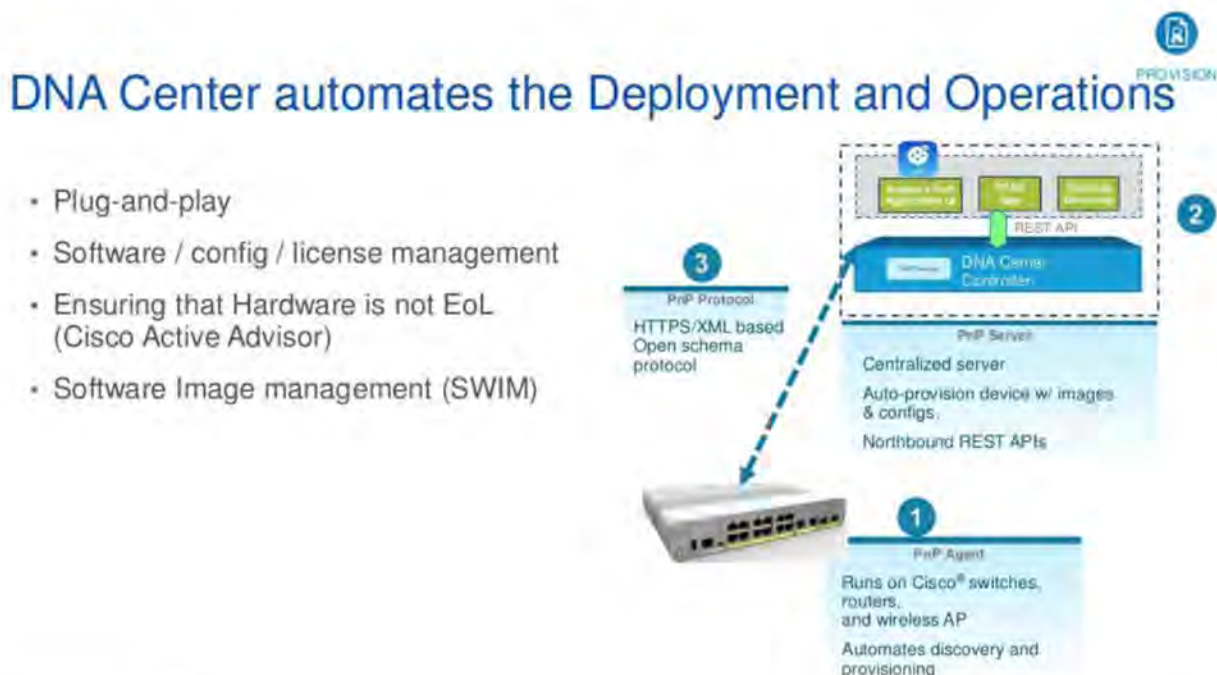


Figure 1

Cisco Connect Toronto 2017 - Introducing the Network Intuitive (Slide 26)

<https://www.slideshare.net/CiscoCanada/cisco-connect-toronto-2017-introducing-the-network-intuitive>

It is the Swarm position that A and B are equivalent. This also disposes of the obfuscating issue of whether the "controller" is mapped to a human.

Second:

Your objection is composed of multiple sub-objections that will be addressed one at a time.

2.1

You said that Cisco's Plug and Play devices could not be claimed to be "co-processors". A co-processor – a device such as a router – is capable of performing network processing tasks as illustrated in the table below.



A Patent '004	B Cisco References
<p>A co-processor – device- is capable of performing any of the following tasks:</p> <p>without limitation: data encryption; graphics, video, and audio processing; direct memory access; mathematical computations; data mining; game algorithms; ethernet packet and other network protocol processing including construction, reception and transmission of data the outside network; financial services and business methods; search engines; internet data streaming and other web-based applications;</p> <p>('004 Patent, Column 4, lines 13-20)</p>	<p>Here is a non-limiting example of the capability of a Cisco device:</p> <p>A router is a networking device that routes data packets between computer networks. A router can connect networked computers to the Internet (https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/network-switch-how.html)</p>

Table 2

2.2

Cisco quotes (mostly from the background of the '004 patent):

The '004 patent explains that that unlike a traditional central processing unit (CPU) that executes machine coded instructions serially, the CPU of the '004 patent uses “[t]rue parallel or multi-core processing” to “partition[] the computational requirements into tasks and distribute[] the tasks to co-processors”. Such parallel processing “divides a large computational requirement into discrete models of executable code.”

Following this, Cisco contrasts the '004 execution using "concurrently executable threads" to the Cisco device operation. There is not, however, a claim requirement of "concurrently executable threads". Cisco seems to be attempting to read background material into the claim.

To further state that "the CPU in a Cisco Plug and Play device operates independently of any other CPU" seems to be an admission to the claim language "without any communication between the first co-processor and the controller".



The CPU described in patent '004 operates *independently* from any other CPU or co-processor, this fact is clearly stated on the patent:

“Those skilled in the art will appreciate that interoperability among the CPU and co-processors may be facilitated by configuring the CPU to compose and/or structure tasks at a level of abstraction which is *independent* of the instruction set architecture associated with the various co-processors, thereby allowing the components to *communicate at a task level* rather than at an instruction level.” (emphasis added, Patent '004, Column 3, lines 34-40)

Furthermore:

“The tasks may be executed *independently* or collaboratively, depending on the task thread restrictions (if any) provided by the CPU.” (emphasis added, Patent '004, Column 3, lines 21-24)

2.3

Cisco states:

Further, a Cisco Plug and Play device does not receive “threads” of “machine coded instructions” from a CPU. As an example, the Configuration Guide describes the use of configuration files to perform configuration upgrades of Cisco Plug and Play devices. Such configuration files contain uncompiled scripts and not “machine coded instructions.”

Moreover, such configuration files are not “partition[ed]” into threads or divided into “discrete models of executable code” and distributed to the Cisco Plug and Play devices. Instead, the configuration files are provided in their entirety to the Cisco Plug and Play devices. For instance, the Configuration Guide describes that “[t]he new configuration



file is copied from the file server to the device through copy command and file check is performed to check the validity of the file.

If the file is valid, then the file is copied to startup configuration.” Thus, the Configuration Guide describes that the entire configuration file is copied to the Cisco Plug and Play device, and not “individual blocks of computations,” “partitions,” or “discrete models of executable code.” As such, the Cisco Plug and Play device processes the entire configuration file on its own and does not “co-process” the configuration file.

Firstly, in Patent ‘004, the terms “thread” and “task” are used interchangeably, i.e., “co-processors configured to proactively retrieve threads (tasks) from the task pool.” (Patent ‘004, Column 2, lines 13-14)

Secondly, your reference to the “discrete modules of executable code” again takes language from the BACKGROUND section of the patent and is not an element of the claim.

It is important to focus on the meaning of a “task” as described in Patent ‘004. As indicted below, the “task” is not limited to “machine coded instructions;” instead, it includes a descriptor which is a data structure which includes the location of the data, i.e. location of the image or configuration file. Again, language from the '004 patent is in column A on the left and descriptions of the Cisco device, from Cisco references are in column B on the left.

A Patent ‘004	B Cisco References
<p>A task 22 may have a task type and a <i>descriptor</i>. (emphasis added, '004 Patent, Column 7, lines 15-16)</p> <p>In an embodiment, the descriptor is a data structure containing a header and a plurality of reference pointers to memory locations, and the task 22 includes the memory address of the data structure. The</p>	<p>Here is an example:</p> <p>When the PnP agent is installed on a high-availability device, once ImageInstall service gets the data structure, the agent determines that the request is for a high-availability device. The active route processor (RP) which is running the PnP agent performs all the operations needed to install the image on both active and standby devices.</p>



<p>header defines the function or instruction to be executed. A first pointer references the location of the data to be processed. A second, optional pointer, references the location for placement of processed data. (emphasis added, '004 Patent, Column 7, lines 47-54)</p>	<p>... 1 Copies the image from the file server to a local disk of the device that is running the active RP. Information about the file server, image location, and destination is populated in the data structure. (Cisco Network Plug and Play Agent Configuration Guide, Cisco IOS XE Everest 16.5.1b, page 8)</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

table 3

2.4

Cisco stated:

As previously explained, the Configuration Guide describes that the Cisco Plug and Play server prepares a “work request” containing the appropriate configuration for the Cisco Plug and Play device based on the device’s unique device identifier (UDI). Since a “work request” is created on-demand and is matched to a single UDI that is executable by a single, unique Cisco Plug and Play device, a “work request” is not executable in parallel by two or more Cisco Plug and Play devices. And because Cisco Plug and Play devices can only execute their own unique “work requests,” there is no “task pool” of “work requests” that are “distribute[d] . . . among two or more” Cisco Plug and Play devices.²⁶

Again, Cisco is attempting to read background material into the claim. Patent ‘004 does not claim that “tasks” (or “work requests”) are distributed among two or more devices. The phrase “distribute[d] . . . among two or more” belongs to the BACKGROUND section of the patent document, not to the claims. In contrast, the SUMMARY OF THE INVENTION describes, “The tasks may be executed *independently* or collaboratively, depending on the task thread restrictions (*if any*) provided by the CPU” (emphasis added, Patent ‘004, Column 3, lines 21-24)

Furthermore, the co-processor described in patent ‘004 is capable of executing a unique and single (not distributed) “work request” associated with the cell:



“In an embodiment, each cell preferably has a *unique*, dedicated agent. In particular, cell 312 includes an agent 320; cell 314 includes an agent 322; cell 316 includes an agent 324; and cell 318 includes an agent 326. *Each agent preferably includes an information field or header which identifies the type of tasks its associated cell is configured to perform*, for example, a *single task*” (emphasis added, Patent ‘004, Column 11, lines 1-7)

Third:

This objection is composed of multiple sub-objections that will be addressed one at a time.

3.1:

Your letter states:

Cisco explained in its Aug. 27, 2019 letter that the Cisco Network Plug and Play Agent is not “configured to . . . retrieve a first task from the task pool” as required by claim 1 of the ’004 patent.

In response, you [Swarm] stated that Cisco’s Plug and Play agent “uses methods like DHCP . . . to acquire the desired IP address of the PnP server”; “relies on DHCP packets, and others, as methods of communication”; that “DHCP packets are analogous to the agent defined in the ’004 patent”; and that “the term agent refers to a software module, analogous to a network packet, associated with a co-processor that interacts with the task pool.”

To the extent that you are alleging that the DHCP packet is an agent, you have not provided any evidence that Cisco’s Plug and Play device uses a DHCP packet to “retrieve a first task from a task pool” as recited in claim 1.

As explained in our September 3 response:

Even though the Configuration Guide does not use the word “agent” to describe how the Plug and Play agent communicates with the PnP server, Cisco does say that the PnP agent relies on DHCP packets, and others, as methods of communication.



To further explain the concept, it is imperative to examine the protocol used by “DHCP packets, and others, as methods of communication.” Such protocol is called IPv4 (Internet Protocol version 4) – and IPv6 with limited support – as noted below:

“**DHCP** Option based Discovery over an **IPv4** Network” (emphasis added, Cisco Network Plug and Play Agent Configuration Guide, page 20)

“The **DNS** server is available as part of the **4G** network and the cloud portal should be programmed to redirect the calling device to an appropriate Cisco Network PnP server for provisioning the device. Currently, Cisco Network PnP support over **4G** interface uses only the **IPv4** network” (emphasis added, Cisco Network Plug and Play Agent Configuration Guide, page 24)

“**Cisco Cloud** Redirection over an **IPv4** and **IPv6** Network” (emphasis added, Cisco Network Plug and Play Agent Configuration Guide, page 23)

“Creates a **HTTP** transport configuration for the PnP agent profile based on the **IPv4** address of the server on which the PnP agent is deployed” (emphasis added, Cisco Network Plug and Play Agent Configuration Guide, page 30)

“A Cisco IOS device that supports **PnP protocol** (that uses HTTP transport)” (emphasis added, Cisco Network Plug and Play Agent Configuration Guide, page 30). By extension, the PnP Protocol operates under an IPv4 network, i.e. HTTP transport.

As cited above, all method of communication supported by the Cisco Plug and Play device — i.e., DHCP, DNS, 4G, Cisco Cloud, HTTP, and PnP Protocol —are based on IPv4. For this reason, instead of limiting the discussion to DHCP, it is essential to examine the common denominator protocol — i.e. IPv4 — used by all Cisco’s Plug and Play device to “retrieve a first task from a task pool.”

Firstly, Table 4 maps the message format as described in Patent ‘004 with the message format as defined by IPv4.

A Patent ‘004 Message Format	B IPv4 Message Format
“an agent is generally analogous to a data frame in the networking sense, in that an agent may be equipped with a source address , a destination address, and a	“The IPv4 datagram is conceptually divided into two pieces: the header and the payload . The header contains addressing and control fields, while the payload carries the actual data to be sent



<p>payload “ ('004 Patent, Column 8, lines 30-33)</p>	<p>over the internetwork.” [IP Version 4 (IPv4) Datagram General Format, page 1]</p> <p>Table 56: [IP Version 4 (IPv4) Datagram General Format, page 1]</p> <p>“Source Address. The 32-bit IP address of the originator of the datagram.</p> <p>Destination Address: The 32-bit IP address of the intended recipient of the datagram” [IP Version 4 (IPv4) Datagram General Format, page 2]</p>
-----------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 4

Secondly, Table 5 demonstrates that the task pool is equipped with the capability to send a “datagram” in response to the “datagram” sent by the cell. Hence, when the cell “retrieves” a task from a task pool, the returning “datagram” carries the task transmitted by the task pool.

A Patent '004	B Cisco References
<p>The communication between the cell and the task pool is bidirectional.</p> <p>“both the cell and the task pool may be equipped with a transceiver.” ('004 Patent, Column 6, lines 22-23)</p>	<p>The communication between the PnP agent and PnP server is bidirectional.</p> <p>“PnP Protocol: Protocol between the PnP agent and PnP server.” (Cisco Network Plug and Play Agent Configuration Guide, page 39)</p>
<p>During the “communication channel” phase, when the task pool transmits the cell receives.</p> <p>“the task pool may answer wirelessly to the cells by equipping the task pool with a transmitter and the solidarity cells with a receiver”</p>	<p>During the “communication channel” phase, when the PnP server transmits the device receives.</p> <p>“The PnP server if it has any task for the device, sends a work request. (Cisco Network Plug and Play Agent Configuration Guide, page 11)</p>



<p>('004 Patent, Column 9, lines 61-63)</p> <p>“During the communication channel phase, the cell receives the task and begins to execute the task.”</p> <p>('004 Patent, Column 6, lines 28-30)</p>	<p>“This secure communication channel is leveraged by the server to send deployment information to the agent.”</p> <p>(Cisco Network Plug and Play Agent Configuration Guide, page 18)</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 5

Thirdly, in the practical terms, the agent sent by the cell “retrieves” a task from the task pool; however, in implementation terms, the interchange is composed of two datagrams, the first datagram is originated by the cell, and the second datagram is originated by the task pool. This is explicitly described in the invention, see Table 6.

A Patent '004	B IPv4 Message Format
<p>As described below, when the cell originates the agent, the destination address is the address of the task pool. Conversely, when the task pool originates the agent, the destination address is the address of the cell.</p> <p>“In an embodiment, the destination address is the address of the task pool 13 when the agent 30 is seeking a task 22, and the destination address is the address of the corresponding cell 12 when the agent 30 is returning to its cell with a task 22.”</p> <p>('004 Patent, Column 8, lines 33-37)</p>	<p>Table 56: [IP Version 4 (IPv4) Datagram General Format, page 1]</p> <p>“Source Address: The 32-bit IP address of the originator of the datagram.</p> <p>Destination Address: The 32-bit IP address of the intended recipient of the datagram.”</p> <p>[IP Version 4 (IPv4) Datagram General Format, page 2]</p>
<p>As described below, when the cell originates the agent, the source address is the address of the cell. Conversely, when the task pool originates the agent, the source address is the address of the task pool.</p>	<p>Table 56: [IP Version 4 (IPv4) Datagram General Format, page 1]</p> <p>“Source Address: The 32-bit IP address of the originator of the datagram.</p>



<p>“Correspondingly, the source address is the address of the cell 12 when the agent 30 is seeking a task 22, and the source address is the address of the task pool 13 when the agent 30 is returning to its cell with a task 22.” (’004 Patent, Column 8, lines 37-40)</p>	<p>Destination Address: The 32-bit IP address of the intended recipient of the datagram.” [IP Version 4 (IPv4) Datagram General Format, page 2]</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------

Table 6

3.2:

You stated:

[T]he Configuration Guide does not describe that Cisco’s Plug and Play device uses a single DHCP packet to both request and receive a configuration.

Patent ’004 does not claim that a single packet it used to request and receive configuration, see Table 6.

3.3:

You also stated:

[T]he Configuration Guide does not describe that a payload containing a “descriptor of the task” is added to a DHCP packet sent by Cisco’s Plug and Play device.

As explained on Table 4, the datagram —i.e. packet — includes a “payload.” Furthermore, as explained on Table 3, the descriptor is a data structure; hence, when the PnP agent gets the data structure, such data structure is transported in the payload section of the datagram.



3.4:

Your letter states:

[T]he Configuration Guide explains that the Cisco Plug and Play agent sends a “request for work” to the Cisco Plug and Play server in a first step. Then, in a second step, the Cisco Plug and Play server sends a “work request” to the Cisco Plug and Play device.

Thus, the Configuration Guide describes that two different types of requests are sent and received—a “request for work” is sent, but a “work request” is received

Patent ‘004 also describes two datagrams, see Table 6. The first datagram is the “request for work” and the second datagram is the received “work request.”

Fourth:

You again stated in your letter:

Cisco explained in its Aug. 27, 2019 letter that your claim chart fails to identify any evidence that a Cisco Plug and Play device delivers a task to itself.

As noted in the response to 3.1: Using practical terms, the agent sent by the cell “retrieves” a task from the task pool, however, in implementation terms, the interchange is composed of two datagrams, the first datagram is originated by the cell, and the second datagram is originated by the task pool; this is explicitly described in the invention, see Table 6.

Accordingly, the patent claim can be interpreted as follows:

“a first co-processor configured to successively: *retrieve [this is the first datagram, which is originated by the cell]* a first task from the task pool; *deliver [this is the second datagram, which is originated by the task pool]* the first task to the first co-processor;”
(emphasis and description added, Patent ‘004 Claim Elements)

You have cited the SETI@home as prior art. Although the reference describes large scale distributed computing, it does not disclose or suggest the invention as claimed.



Without distracting from Swarm's contention that the '004 patent is relevant to Cisco products, Swarm would like you to consider the relevance of Swarm Patent 9,146,777. A claim chart showing the relationship of the '777 patent to Cisco product is appended below.

U.S. Patent 9,146,777

Parallel processing with solidarity cells by proactively retrieving from a task pool a matching task for the solidarity cell to process

References

[Reference 1] Cisco Network Plug and Play Agent Configuration Guide, Cisco IOS XE Everest 16.6

[Reference 2] Cisco FindIT Network Manager and Probe Administration Guide, Version 2.0

[Reference 3] Cisco Network Plug and Play Solution Guide for SMB

[Reference 4] DHCP IP Version 4 (IPv4) Datagram General Format

[Reference 5] Cisco Understanding Virtual Network Function Descriptors

[Reference 6] Cisco Managing Configuration Files

[Reference 7] Cisco Tech Talks: FindIT PnP Overview and Configuration

Cisco FindIT Network Manager - Device Support List

Routers

RV160/RV160W, RV260/RV260x, RV320/RV325, RV340/RV340W, RV345/RV345P



Switches

200 Series Smart, 250 Series Smart, 300 Series Managed, 350 Series Managed,
500 Series Stackable 550 Series Stackable

SD-Access Wireless

WAP125, WAP150, WAP361, WAP371, WAP571/WAP571E, WAP581

Patent '777 Claim Elements	Cisco devices
<p>1. An apparatus for parallel processing of a large computing requirement, the apparatus comprising: a central processing unit ("CPU");</p>	<p>A non-limiting example of the apparatus is depicted on Appendix A.</p> <p>To run FindIT Network Manager under Ubuntu Linux operating system, your environment must meet the following requirements:</p> <ul style="list-style-type: none"> • Ubuntu version 16.04.x (Xenial Xerus) • CPU: 1x 64-bit Intel architecture <p>[Reference 2, page 4 and 5]</p>
<p>a task pool in electronic communication with the CPU;</p>	<p>FindIT Network Manager includes an embedded Network Plug and Play server. [Reference 3, page 5]</p> <p>The Cisco Network Plug and Play solution for SMBs includes the following components:</p> <ul style="list-style-type: none"> • Cisco FindIT Network Manager ... • Cisco Network Plug and Play server - This embedded application receives Network Plug and Play requests from Cisco devices and provisions devices based on predefined rules and criteria. <p>[Reference 3, page 5]</p>



<p>and a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process;</p>	<p>The Cisco device, having PnP agent contacts the PnP server requesting for a task [Reference 1, page 11]</p> <p>The Cisco Network Plug and Play agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. Once a server is found and connection established, the agent performs deployment related activities like configuration, image, license, and file updates by communicating with the server. [Reference 1, page 9]</p>
<p>wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks;</p>	<p>By virtue of running the FindIT Network Manager, the CPU populates the PnP Sever with one or more matching tasks to be executed by the PnP Enabled Devices.</p> <p>Here is an example of how the FindIt Manager populates tasks to the PnP server:</p> <p>In the [FindIT Network Manager] menu, we'll enter our product ID into the product ID box then click Add New, now we'll select the device from the list at the bottom of the page and click the edit icon in the top left corner once again we'll add our image and configuration files from the image and configure drop-down menus respectively then save by clicking the Save icon in the upper left corner, the network PnP Server will use these files to provision any device with a matching product ID attempting to connect to the network. [Reference 7, page 3]</p> <p>Deployment related operational tasks:</p> <ul style="list-style-type: none"> • Establishing initial network connectivity for the device • Delivering device configuration • Delivering software and firmware images • Delivering licenses • Delivering deployment script files • Provisioning local credentials • Notifying other management systems about deployment related



	<p>events [Reference 1, page 5]</p>
<p>wherein each task comprises a descriptor, the descriptor containing at least: a function to be executed;</p>	<p>The Virtual Network Function Descriptor (VNFD) file describes the instantiation parameters and operational behaviors of the VNFs. [Reference 5, page 1]</p> <p>Deployment related operational tasks:</p> <ul style="list-style-type: none"> • Establishing initial network connectivity for the device • Delivering device configuration • Delivering software and firmware images • Delivering licenses • Delivering deployment script files • Provisioning local credentials • Notifying other management systems about deployment related events <p>[Reference 1, page 5]</p>
<p>and a memory location of data upon which the function is to be executed;</p>	<p>Configuration files are stored in the following locations:</p> <ul style="list-style-type: none"> • The running configuration is stored in RAM (memory) <p>[Reference 6, page 3]</p> <p>The PnP file server hosts files that can be copied over by the deploying devices in the network. The file server can be a dedicated server hosting files or a part of the device hosting the PnP server. [Reference 1, page 7]</p> <p>When the PnP agent receives the work request, executes the task and sends back a reply to the PnP server about the task status, whether it is a success or error, and the corresponding information requested. [Reference 1, page 11]</p>
<p>wherein the first agent is a data frame comprising: a source address, a destination address and a payload;</p>	<p>Both IPv4 and IPv6 addresses can be used for PnP server IP configuration. [Reference 1, page 30]</p>



	<p>Creates a HTTP transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed. [Reference 1, page 31]</p> <p>The IPv4 datagram is conceptually divided into two pieces: the header and the payload. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. [Reference 4, page 1]</p> <p>Table 56: Internet Protocol Version 4 (IPv4) Datagram Format Source Address: The 32-bit IP address of the originator of the datagram. Destination Address: The 32-bit IP address of the intended recipient of the datagram [Reference 4, page 2]</p>
<p>wherein the first agent retrieves the matching task by: being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform;</p>	<p>The Cisco device, having PnP agent contacts the PnP server requesting for a task [Reference 1, page 11]</p> <p>Network Plug and Play enabled devices are used to map individual devices to the desired image and configuration file. Devices are identified by the combination of product ID (PID) and serial number. [Reference 3, 12]</p> <p>Table 56: Internet Protocol Version 4 (IPv4) Datagram Format Source Address: The 32-bit IP address of the originator of the datagram. Destination Address: The 32-bit IP address of the intended recipient of the datagram [Reference 4, page 2]</p>
<p>searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform;</p>	<p>The PnP agent first tries to discover a PnP server, with which it can communicate. Once a server is found and connection established, the agent performs deployment related activities like configuration, image, license, and file updates by communicating with the server. [Reference 1, page 9]</p>

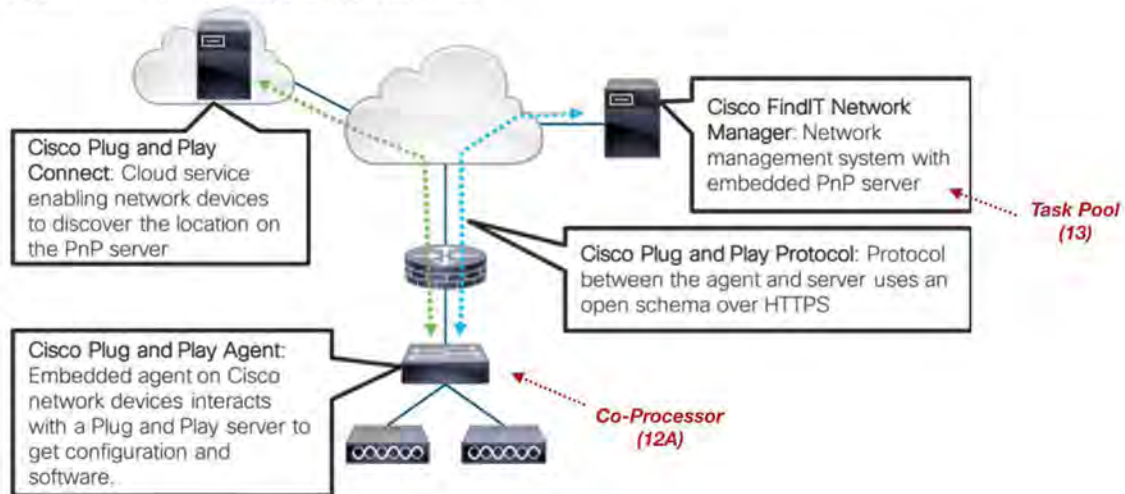


	<p>the PnP-enabled device records are searched, and, if a match is found, the image and configuration files specified will be pushed out to the device. [Reference 3, 12]</p>
<p>and returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task.</p>	<p>if a match is found, the image and configuration files specified will be pushed out to the device. [Reference 3, 12]</p> <p>When the PnP agent successfully acquires the IP address, it initiates a long lived, bidirectional layer 3 connection with the server and waits for a message from the server. The PnP server application sends messages* to the agent requesting for information and services to be performed on the device. [Reference 1, page 5]</p> <p>Table 56: Internet Protocol Version 4 (IPv4) Datagram Format Source Address: The 32-bit IP address of the originator of the datagram. Destination Address: The 32-bit IP address of the intended recipient of the datagram [Reference 4, page 2]</p> <p>*According to IPv4, when the message is sent by the server, the source address is PnP server's address and the destination address is the device's address.</p>



Appendix A

Figure 1. Cisco Network Plug and Play Architecture



Apparatus

This figure is shown in Reference 3, page 5.

Note: The terms *Task Pool*, *Co-Processor* and *Apparatus* were added to associate with the terminology used by Patent '777.



Swarm continues to believe that Swarm Patent 9,852,004 and now also Patent 9,146,777 are relevant to products offered by Cisco. Swarm remains willing to enter into licensing negotiations for a broad license under the Swarm patents.

Sincerely yours,

John A. Fisher
IP licensing Consultant
Cc: Alfonso Iñiguez

From: John Fisher <phxfish@gmail.com>
Sent: Wed, 27 Nov 2019 13:34:54 -0700
Subject: Fwd: Swarm Technology Licensing Opportunity
To: Alfonso Íñiguez <alfonso@swarmtechnology.us>
[Cisco Response to Swarm's 10-29-2019 Letter \(11-27-2019\).pdf](#)

Latest from Cisco. I haven't studied it in detail yet. They have not cited any new art. I have your new '777 claim chart, but am still working my way through the references.

John

----- Forwarded message -----

From: Foster, Theo <Theo.Foster@haynesboone.com>
Date: Wed, Nov 27, 2019 at 12:12 PM
Subject: RE: Swarm Technology Licensing Opportunity
To: John Fisher <phxfish@gmail.com>

Mr. Fisher,

Attached is Cisco's letter response to your letter of October 29, 2019.

Best regards and Happy Thanksgiving,

Theo Foster

Partner
(t) 972.739.8649

From: John Fisher <phxfish@gmail.com>
Sent: Tuesday, October 29, 2019 5:29 PM
To: Foster, Theo <Theo.Foster@haynesboone.com>
Cc: Conner, Gayle <Gayle.Conner@haynesboone.com>; Alfonso Íñiguez <alfonso@swarmtechnology.us>
Subject: Swarm Technology Licensing Opportunity

I have attached a letter in response to your letter of October 11, 2019.

John Fisher

Virus-free. www.avast.com

CONFIDENTIALITY NOTICE: This electronic mail transmission is confidential,

EXHIBIT

Iniguez 20 5/14/21 DJ

exhibitsticker.com

may be privileged and should be read or retained only by the intended recipient. If you have received this transmission in error, please immediately notify the sender and delete it from your system.

haynesboone

November 27, 2019

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

I write in response to your letter dated October 29, 2019.¹ Your letter raised various points and issues with respect to two patents, U.S. Patent No. 9,852,004 (the "'004 patent") and U.S. Patent No. 9,146,777 (the "'777 patent"). I address each in turn below.

'004 Patent

Your October 29 letter continues to allege that the claimed "controller" corresponds to an "administrator for the PnP server,"² that is, a human being. Cisco explained in its August 27, 2019 and October 11, 2019 letters why your mapping of the claimed "controller" to a human administrator is flawed, unsupported, and contrary to the disclosure of the '004 patent.

As an alternative position, your October 29 letter also identifies the claimed "controller" as a Cisco DNA Center controller.³ This new mapping of the claimed "controller" is equally flawed, unsupported, and contrary to the claim language for the following reasons. First, Swarm previously identified the claimed "task pool" as a Plug and Play (PnP) server, which is "a service running on Cisco DNA Center."⁴ Thus, when PnP devices communicate with the PnP Server software, they are also communicating with the Cisco DNA Center controller hardware. Accordingly, Cisco PnP configuration technology does not infringe the '004 patent because

¹ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Oct. 29, 2019) ("Fisher's October 29 Letter")

² Fisher's October 29 Letter at 1.

³ Fisher's October 29 Letter at 2-3.

⁴ Adam Radford, Cisco DNA Center Plug and Play (PnP) – Part 1, <https://blogs.cisco.com/developer/cisco-dna-center-plug-and-play-pnp-part-1> (Jul. 8, 2019).

Haynes and Boone, LLP
Attorneys and Counselors
2505 North Plano Road, Suite 4000
Richardson, TX 75082
Phone: 214.651.5000

Letter to John A. Fisher
November 27, 2019

Cisco's PnP devices do not "retrieve a first task from the task pool . . . without any communication between the first co-processor and the controller," as recited in claim 1.

Similarly, a new PnP device on a network will initiate communication with the PnP server software running on the DNA Center hardware,⁵ *i.e.* the alleged "controller." Thus, Cisco's PnP configuration technology does not "dynamically accept" a new device "into the processing system on a plug-and-play basis without any communication with the controller," as recited in claim 1.

The "without any communication" limitations are wholly absent from Cisco's PnP configuration technology, so there can be no infringement of the '004 patent's claim 1. While we continue to disagree with other arguments and statements made in your October 29 letter, we will not belabor those points in view of the unambiguous distinction between the '004 patent and Cisco's PnP configuration technology.

'777 Patent

The newly provided '777 claim chart purports to map the claim 1 of the '777 patent to the PnP configuration technology. We have reviewed this claim chart carefully, and we note below multiple deficiencies in its analysis.

The '777 patent claims a "solidarity cell" including "a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process." In prosecution, Swarm emphasized that the claimed agent is an "autonomous representative of its corresponding solidarity cell."⁶

Cisco's PnP Devices do not have any such "agent." As discussed previously in Cisco's October 11, 2019 letter, a Cisco Plug and Play agent sends a "request for work" to the Cisco Plug and Play server in a first step. The request for work is not autonomous and does not, itself, search for a task; rather, the Cisco PnP Server software performs this search. As noted above, the PnP Server software runs on a computer that includes a CPU, such as the DNA Center controller. In a network employing the FindIT Network Manager, the PnP Server software executes on the same computer as the FindIT Network Manager.⁷ Thus, even to the extent that Swarm might attempt to assert that a "request for work" corresponds to the claimed "first agent," the computer

⁵ See, e.g., Cisco Connect Toronto 2017 - Introducing the Network Intuitive (Slide 26), <https://www.slideshare.net/CiscoCanada/cisco-connect-toronto-2017-introducing-the-network-intuitive> (showing that DNA Center is involved in enabling "plug-and-play").

⁶ '777 File History, Response at 10 (May 29, 2015).

⁷ Cisco FindIT Network Manager Administration Guide, Version 1.1.x, p. 11, https://www.cisco.com/c/dam/en/us/td/docs/cloud-systems-management/network-automation-and-management/cisco-findit-network-management/admin_guide/b_Cisco_FindIT_Network_Manager_BookMap_1_1_x.pdf ("FindIT Network Manager provides a Cisco Network Plug and Play server...").

Letter to John A. Fisher
November 27, 2019

processor that executes instructions in response to receipt of the request for work is the same computer processor that executes the FindIT Network Manager (which Swarm identified as the alleged “CPU”). Thus, the request for work is incapable of searching for and retrieving a task “without requiring an instruction from the CPU” as claimed.

The ’777 claims also require that the agent be “dispatched,” performing “searching,” and then “return[]” with a matching task. In contrast, and as previously explained in Cisco’s Oct. 11, 2019 letter, the “request for work” sent by a PnP device and the corresponding “work request” sent by the PnP Server are two separate communications. The analysis provided by Swarm does identify an “agent” that performs all three steps in the context of Cisco’s PnP system.

Claim 1 also recites that “the first agent is a data frame” whose “payload comprises a list of functions the first solidarity cell is configured to perform.” The analysis provided by Swarm does not identify any evidence showing that a data frame sent from a PnP Device (the alleged “solidarity cell”) to the PnP Server (the alleged “CPU”) contains a list of functions that the PnP Device is configured to perform. The cited evidence—that a PnP Device provides its product ID and serial number—is plainly not a “list of functions.”

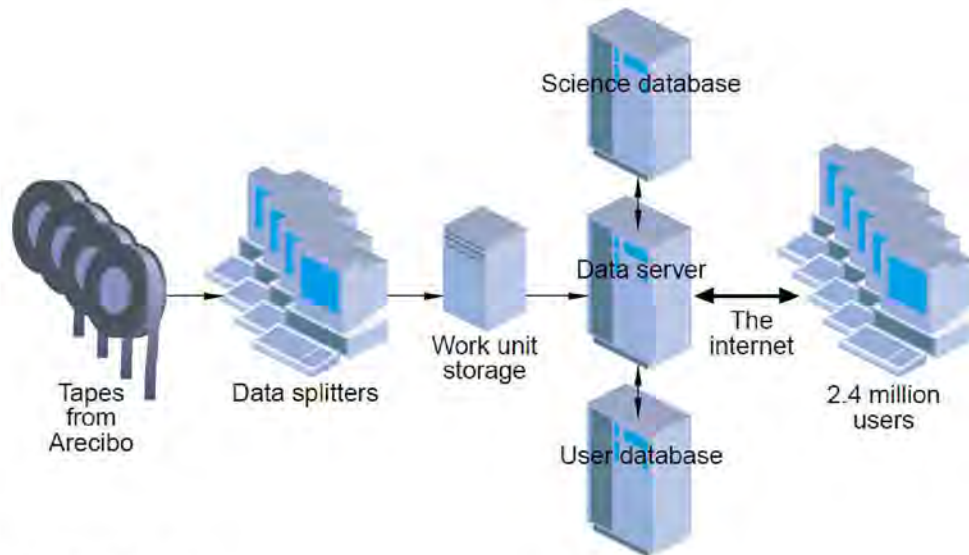
The distinctions discussed above are merely exemplary, but they are more than sufficient to demonstrate that the ’777 patent is not relevant to Cisco’s PnP configuration technology.

Invalidity

As you requested, we previously provided a copy of the prior art article “SETI@Home—Massively Distributed Computing for SETI” for Swarm’s review. We believe that SETI@Home is highly relevant to the asserted claims of both the ’004 and ’777 patents. Together with the background knowledge of a person of ordinary skill in the art, the article teaches or renders obvious the asserted claims as Swarm interprets them.

Swarm has not identified any limitation of the asserted patent claims that is not disclosed or rendered obvious by SETI@Home. The article describes the configuration of numerous home computers to work collectively to analyze satellite observations. As the figure from page 80 shows below, satellite recordings are subdivided into small work units and placed into “work unit storage” by “data splitters.” More than two million users’ computers then communicate with the work unit storage via a data server; but the users’ computers do not communicate with the data splitters. Communications between the users’ computers and the server is over the Internet via HTTP. Anyone who was interested in joining the processing collective could do so by downloading software from the SETI@home website; the software would then install itself.

Letter to John A. Fisher
November 27, 2019



SETI@Home, Fig. 2 at 80.

Based on Swarm's interpretation of the claims, we do not see any basis for Swarm's conclusory statement that SETI@Home "does not disclose or suggest the invention as claimed."⁸ In the event that Swarm persists in its allegations against Cisco, we ask the Swarm provide a detailed response to the prior art.

Conclusion

In summary, we have carefully considered and responded to your concerns regarding the '004 and '777 patents. For the reasons established above and in the parties' prior communications, we believe that a license would not be of benefit to Cisco. Accordingly, we are prepared to close this matter, and we will do so absent further communication from Swarm. Please direct any further communications to my attention.

Very truly yours,

Theo Foster
(972) 739-8649 – Phone
theo.foster@haynesboone.com

⁸ Fisher's October 29 Letter at 13.

From: John Fisher <phxfish@gmail.com>
Sent: Fri, 13 Dec 2019 16:23:10 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: Alfonso Íñiguez <alfonso@swarmtechnology.us>, "Conner, Gayle" <gayle.conner@haynesboone.com>
[cisco 121319.pdf](#)

Dear Mr. Foster,
I have attached a letter in response to your letter of November 27, 2019.
John

EXHIBIT

Iniguez 21 5/14/21 DJ

exhibitsticker.com



December 13, 2019

Via US Mail and email to theo.foster@haynesboone.com

Theo Foster
Haynes and Boone, LLP
2505 North Plano Road, Suite 4000
Richardson, TX 75082

Re: Swarm Technology Licensing Opportunity FRE 408

Dear Mr. Foster:

In your letter of November 27, 2019 you raised several issues with respect to Swarm's application of Swarm's '004 and '777 patents to Cisco products. This letter addresses those issues. Swarm relies on six references:

[Reference 1] Cisco Network Plug and Play Agent Configuration Guide
[Reference 2] Getting Started with Cisco DNA Center
[Reference 3] Network Plug and Play Solution Guide for SMB
Reference 4] SETI@HOME—MASSIVELYDISTRIBUTED COMPUTING FOR SETI
Reference 5] BOINC: A System for Public-Resource Computing and Storage
Reference 6] Folding@home - The Grid Compute Architecture

The first issue, with respect to the '004 patent, can be broken into two parts:

Issue 1.1:

'004 Patent

Your October 29 letter continues to allege that the claimed "controller" corresponds to an "administrator for the PnP server," that is, a human being. Cisco explained in its August 27, 2019 and October 11, 2019 letters why your mapping of the claimed "controller" to a human administrator is flawed, unsupported, and contrary to the disclosure of the '004 patent.

Response:

As stated in my October 29, 2019 letter:

The administrator for the PnP server **sets** device authentication mechanisms which are acceptable for a particular deployment.
(emphasis added, Reference 1, page 18)

(480) 319-2233

phxfish@gmail.com

8300 S. Homestead Lane, Tempe, AZ 85284



Besides “setting” the device authentication mechanisms for a particular deployment, the administrator also “pre-provisions” the PnP server with deployment operational tasks to be used by the devices, see Figure 1. Such deployment tasks include:

Deployment related operational tasks:

- Establishing initial network connectivity for the device
- Delivering device configuration
- Delivering software and firmware images
- Delivering licenses
- Delivering deployment script files
- Provisioning local credentials
- Notifying other management systems about deployment related events

(Reference 1, page 5)

As illustrated in Figure 1, the “Plug and Play application” is pre-provisioned by the network administrator. The terms PnP server and Plug and Play application are interchangeably used in Reference 1. The interchangeability of terms is reasonable since “The PnP server is an **application** running as the network manager” (emphasis added, Reference1, page 17).

The concept of running the PnP server as an application is foreseen in Patent '004, “the task pool 13 may be software based” (Patent '004, column 7, line 4-5), and reiterated here: “the task pool may be in the form an external device or **application** located within wireless reach from the laptop” (emphasis added, Patent '004, column 12, lines 8-10).

The crucial question is:

What mechanism does the administrator use to enter the pre-provisioning information into the PnP server?

The administrator uses a “deployment application” running on a smartphone or PC to enter the pre-provisioning information into PnP server — “The PnP server also communicates with proxy servers like **deployment applications** on smart phones and PCs” (Reference 1, page 10).

Thus the “deployment application running on their PC or on a smart phone” is used to pre-provision the PnP server because the same can be alternatively used to manually configure the devices, “A Cisco device can alternatively be manually configured by the network administrator using a **deployment application** running on their PC or on a smart phone.” (emphasis added, Reference 1, page 17).

Given that the administrator uses a smartphone or PC to enter the enter the information into the PnP server, it follows that the smartphone or PC is the “controller” that includes a user interface, as described in Patent '004:



"the **controller** 402 may be a **smartphone**, tablet, **laptop**, or other device which may include a display 404 and a **user** interface (e.g., keypad) 406 for facilitating user interaction with the various devices on the network." (emphasis added, '004 Patent, Column 11, lines 46-50)

Figure 1: Simplified Deployment Server

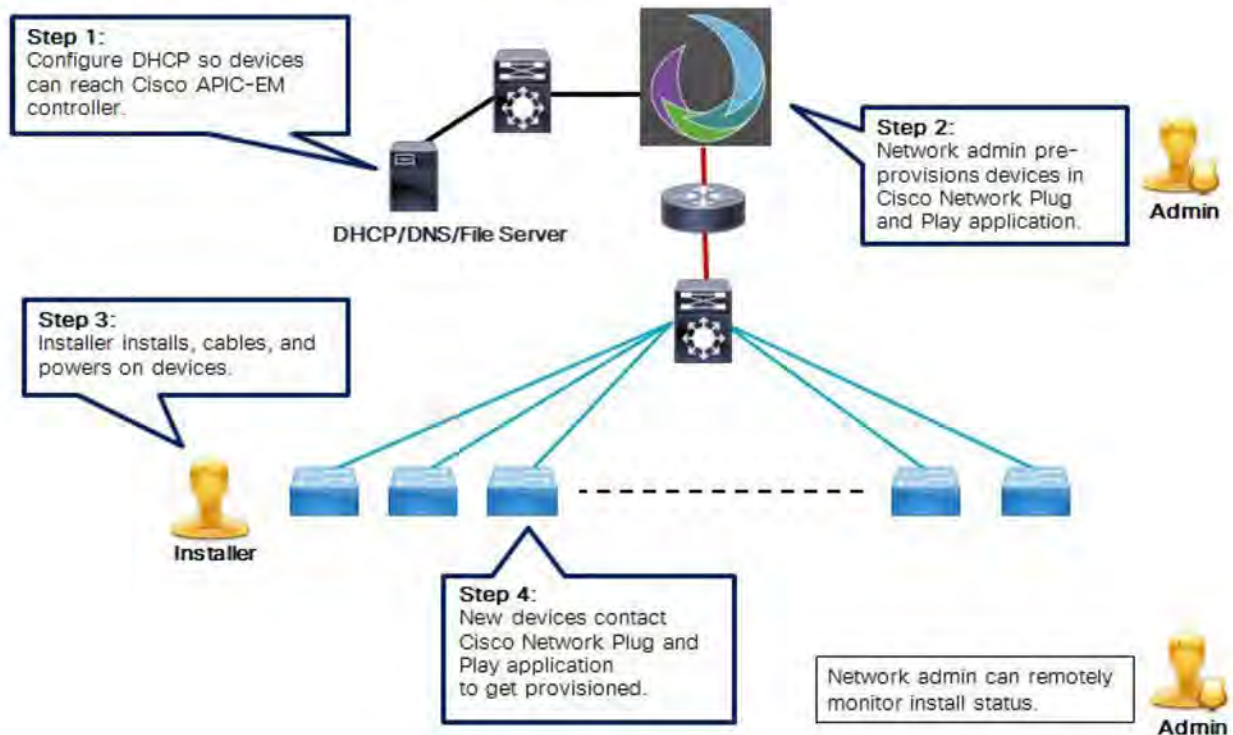


Figure 1. Cisco Network Plug and Play Agent Configuration Guide, page 10.

Issue 1.2:

'004 Patent

As an alternative position, your October 29 letter also identifies the claimed "controller" as a Cisco DNA Center controller. This new mapping of the claimed "controller" is equally flawed, unsupported, and contrary to the claim language for the following reasons. First, Swarm previously identified the claimed "task pool" as a Plug and Play (PnP) server, which is "a service running on Cisco DNA Center." Thus, when PnP devices communicate with the PnP Server software, they are also communicating with the Cisco DNA Center controller hardware.

Accordingly, Cisco PnP configuration technology does not infringe the '004 patent because Cisco's PnP devices do not "retrieve a first task from the task pool . . . without any communication between the first co-processor and the controller," as recited in claim 1.



Similarly, a new PnP device on a network will initiate communication with the PnP server software running on the DNA Center hardware, i.e. the alleged "controller." Thus, Cisco's PnP configuration technology does not "dynamically accept" a new device "into the processing system on a plug-and-play basis without any communication with the controller," as recited in claim 1.

The "without any communication" limitations are wholly absent from Cisco's PnP configuration technology, so there can be no infringement of the '004 patent's claim 1. While we continue to disagree with other arguments and statements made in your October 29 letter, we will not belabor those points in view of the unambiguous distinction between the '004 patent and Cisco's PnP configuration technology.

Response:

The concept of running the PnP server as software is preemptively proposed in Patent '004 which describes "the task pool 13 may be software based" (Patent '004, column 7, line 4-5). Hence, it is fitting for the cell to communicate with the task pool application without being instructed to do so by the controller, as described below:

"the co-processors are referred to as autonomous, proactive solidarity cells. In this context, the term autonomous implies that a **co-processor may interact with the task pool without being instructed to do so by the CPU**" (emphasis added, Patent '004, column 2, lines 36-40)

Hence, even if the task pool is software-based and the software is executed by the CPU ("controller"), the co-processors interact with the task pool without being instructed to do so by the CPU.

This argument is additionally fortified by demonstrating that multiple "controllers" may populate the task pool as well. In such case, the alternative "controllers" are indisputably dissociated from the PnP server application.

Consider the following: Patent '004 describes that "multiple CPUs 11 may share one or more task pools" (Patent '004, column 4, lines 52-53). Such is the case of the Cisco DNA Center's PnP Server which is also populated by detached controllers:

"Access DNA Center by entering its network IP address in your browser." (Reference 2, page 1)

Moreover, upon logging into the DNA Center by using any of the supported browsers: "Google Chrome, version 62.0 or later, Mozilla Firefox, version 54.0 or later" (Reference 2, page 1), the administrator uses the detached "controller" to populate the Cisco DNA Center PnP server:



"After you log in to DNA Center, you are taken to the DNA Center home page, which is divided into two main areas—Applications and Tools:

Applications include:

...

- **Policy**—Create policies that reflect your organization's business intent for a particular aspect of the network, such as network access. DNA Center takes the information collected in a policy and translates it into network-specific and device-specific configurations required by the different types, makes, models, operating systems, roles, and resource constraints of your network devices.
 - **Provision**—Prepare and configure devices, including adding devices to sites, assigning devices to the DNA Center inventory, deploying the required settings and policies, creating fabric domains, and adding devices to the fabric."
- (Reference 2, page 2)

The second issue, with respect to the '777 patent, can be broken into four parts:

Issue 2.1:

The '777 patent claims a "solidarity cell" including "a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process." In prosecution, Swarm emphasized that the claimed agent is an "autonomous representative of its corresponding solidarity cell."

Cisco's PnP Devices do not have any such "agent." As discussed previously in Cisco's October 11, 2019 letter, a Cisco Plug and Play agent sends a "request for work" to the Cisco Plug and Play server in a first step. The request for work is not autonomous and does not, itself, search for a task; rather, the Cisco PnP Server software performs this search.

Response:

Even though the *Cisco Network Plug and Play Agent Configuration* Guide does not use the term "agent" in reference to the communication protocol between the Plug and Play Agent and the PnP Server, the documentation does say that the PnP Agent communication is "based on the IPv4 address of the server on which the PnP Agent is deployed. (Reference 1, page 31).

In such IPv4 protocol, when requesting a task from the PnP Server, the PnP Agent sends an IPv4 datagram where the source address is the PnP Agent's address and the destination address is the PnP Server's address. Thereafter, when the task is provided by the PnP Server, the source address is the PnP Server's address and the destination address is the PnP Agent's address. That is, the transaction consists of two datagrams, the first datagram is originated by the PnP Agent, and the second datagram is originated by PnP Server.

Correspondingly, as described in Patent '777, the agent is "dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the



destination address is the task pool's address" (Patent '777, Claim 1). Thereafter, when the task is provided by the task pool, "returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address." (Patent '777, Claim 1). That is, the transaction consists of two datagrams, the first datagram is originated by the cell, and the second datagram is originated by the task pool in the same manner as in the Cisco device.

Furthermore, the request for work is indeed autonomous, as described below:

"any Cisco device *automates* the following deployment related operational tasks:

- Establishing initial network connectivity for the device
 - Delivering device configuration
 - Delivering software and firmware images
 - Delivering licenses
 - Delivering deployment script files
 - Provisioning local credentials
 - Notifying other management systems about deployment related events
- (emphasis added, Reference 1, page 5)

Lastly, the Cisco PnP Server software performs the search only after receiving the request from the PnP Agent. The PnP Server will not search itself without a request for the PnP Agent. The important matter to grasp is that the requesting PnP Agent is the one that triggers the assignment, even if the search is done the PnP Server. In the exact same manner, in Patent '777, the requesting cell is the one that triggers the assignment, "The task pool 13 may use the prioritization table *to determine which of the eligible tasks 22 to assign to a requesting cell 12A*" (emphasis added, Patent '777, column 4, lines 37-39).

Issue 2.2:

the PnP Server software runs on a computer that includes a CPU, such as the DNA Center controller. In a network employing the FindIT Network Manager, the PnP Server software executes on the same computer as the FindIT Network Manager. Thus, even to the extent that Swarm might attempt to assert that a "request for work" corresponds to the claimed "first agent," the computer processor that executes instructions in response to receipt of the request for work is the same computer processor that executes the FindIT Network Manager (which Swarm identified as the alleged "CPU"). Thus, the request for work is incapable of searching for and retrieving a task "without requiring an instruction from the CPU" as claimed.

Response:

This objection was addressed in the response to Issue 1.2.



Issue 2.3:

The '777 claims also require that the agent be "dispatched," performing "searching," and then "return[]" with a matching task. In contrast, and as previously explained in Cisco's Oct. 11, 2019 letter, the "request for work" sent by a PnP device and the corresponding "work request" sent by the PnP Server are two separate communications. The analysis provided by Swarm does identify an "agent" that performs all three steps in the context of Cisco's PnP system.

Response:

This issue was addressed in the response to Issue 2.1.

Issue 2.4:

Claim 1 also recites that "the first agent is a data frame" whose "payload comprises a list of functions the first solidarity cell is configured to perform." The analysis provided by Swarm does not identify any evidence showing that a data frame sent from a PnP Device (the alleged "solidarity cell") to the PnP Server (the alleged "CPU") contains a list of functions that the PnP Device is configured to perform. The cited evidence—that a PnP Device provides its product ID and serial number—is plainly not a "list of functions."

Response:

First, the terms "list of functions" and "types of tasks" are interchangeably used in Patent '777.

Compare:

"the payload comprises a **list of functions** the first solidarity cell is configured to perform;" (emphasis added, Patent '777, claim 1)

With:

"the payload contains identifying information of the **types of tasks** the corresponding cell 12A... n can perform" (emphasis added, Patent '777, column 5, lines 56-57)

Second, individual devices are mapped to desired image and configuration files:

"Network Plug and Play enabled devices are used to **map** individual devices to the desired image and configuration file. Devices are identified by the combination of product ID (PID) and serial number" (emphasis added, Reference 3, page 12)

Third, the device's product ID (PID) and serial number information is transmitted by the PnP Agent when the "PnP agent contacts the PnP server requesting for a task (Reference 1, page 11). In this manner:

"The PnP agent provides capability to extract device inventory and other important information to the PnP server on request" (Reference 1, page 7)



Forth, the device's product ID (PID) and serial number information is used by the PnP Server to assign any of the following tasks:

"any Cisco device automates the following deployment related operational **tasks**:

- Establishing initial network connectivity for the device
- Delivering device configuration
- Delivering software and firmware images
- Delivering licenses
- Delivering deployment script files
- Provisioning local credentials

(emphasis added, Reference 1, page 5)

Fifth, Patent '777 considers the case in which cells are capable of executing only one task of a unique type:

"For example, in FIG. 1, the system 10 divides a computing problem into tasks of a first type, a second type, and a third type; a first cell 12A is capable of performing **only tasks of the first type**; a second cell 12B can perform tasks of the second type; a third cell 12C can perform tasks of the third type;" (emphasis added, Patent '777, column 3, lines 62-67)

In conclusion, according the Patent '777, it is possible to have a "list of functions" that includes only a task of the first type that the corresponding cell is capable of performing. Such is the case of the PnP Agent which sends its product ID (PID) and serial number to the PnP Server; in turn the PnP Server uses the product ID (PID) and serial number as a capability identifier to assign a task to the device.

Issue 3:

As you requested, we previously provided a copy of the prior art article "SETI@Home—Massively Distributed Computing for SETI" for Swarm's review. We believe that SETI@Home is highly relevant to the asserted claims of both the '004 and '777 patents. Together with the background knowledge of a person of ordinary skill in the art, the article teaches or renders obvious the asserted claims as Swarm interprets them.

Swarm has not identified any limitation of the asserted patent claims that is not disclosed or rendered obvious by SETI@Home. The article describes the configuration of numerous home computers to work collectively to analyze satellite observations. As the figure from page 80 shows below, satellite recordings are subdivided into small work units and placed into "work unit storage" by "data splitters." More than two million users' computers then communicate with the work unit storage via a data server; but the users' computers do not communicate with the data splitters. Communications between the users' computers and the server is over the Internet via HTTP. Anyone who was interested in joining the processing collective could do so by downloading software from the SETI@home website; the software would then install itself.

SETI@Home, Fig. 2 at 80.



Based on Swarm's interpretation of the claims, we do not see any basis for Swarm's conclusory statement that SETI@Home "does not disclose or suggest the invention as claimed."⁸ In the event that Swarm persists in its allegations against Cisco, we ask the Swarm provide a detailed response to the prior art.

Response:

The article that you have provided offers insights into the software system used by SETI@home.

One of the authors of the article is David Anderson who is the SETI@home project director:

"SETI@HOME—MASSIVELY DISTRIBUTED COMPUTING FOR SETI
By Eric Korpela, Dan Werthimer, **David Anderson**, Jeff Cobb, and Matt Lebofsky"
(emphasis added, Reference 4, page 78)

and

"**David Anderson** is the *SETI@home project director*." (emphasis added, Reference 4, page 82)

Furthermore, the article instructs the user to download the SETI@home client program from:
(<http://setiathome.ssl.berkeley.edu>) (Reference 4, page 80)

Upon navigating the website mentioned above, we can see that the computer architecture is
"Powered by BOINC."

A paper titled, "BOINC: A System for Public-Resource Computing and Storage," which is
coincidentally written by SETI@home project director David Anderson, explains that
SETI@home is not the only project that uses BOINC:

"Projects using BOINC

A number of public-resource computing projects are using BOINC. The requirements of
these projects have shaped the design of BOINC.

SETI@home, a continuation of the original SETI@home project [1], performs digital
signal processing of radio telescope data from the Arecibo radio observatory. A BOINC-
based version of this project has been developed, and we are currently shifting the
existing SETI@home user base (over 500,000 active participants) to the BOINC-based
version. The BOINC based SETI@home will use client disks to archive data, eliminating
the need for its central tape archive.

Predictor@home: [11] This project, based at The Scripps Research Institute, studies
protein behavior using CHARMM, a FORTRAN program for macromolecular dynamics
and mechanics. It is operational within Scripps, and is being readied for a public launch.

Folding@home [10]. This project is based at Stanford University. It studies protein
folding, misfolding, aggregation, and related diseases. It uses novel computational



methods and distributed computing to simulate time scales thousands to millions of times longer than previously achieved.” (emphasis added, Reference 5, section 2.2)

From the projects listed above, Folding@home is particularly notable because its superbly documented on GitHub. Consider the following:

“The Grid Compute Architecture

Overview

This is an implementation of the GComp architecture. It's composed of three parts:

- The *Project Management Server*
- The *Job Queue Server*
- The *Job Client*

These three components form a powerful architecture for management of execution tasks that are to be handled in a distributed cloud environment. Below follows an introduction to these components and an overview of how jobs are created and sent out to clients.

Project Management Server

The *project management server* interacts with the *job server to upload job requests to a queue*. In turn, the *job server is contacted by clients who pop elements from the queue and execute the request by downloading a script from the supplied server in the request*.” (emphasis added, Reference 6, page 1)

As we can see from the description above, “*the project management server interacts with the job server to upload job requests to a queue*,” similarly, “*a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks*,” (Patent '004, claim 1).

Hence, the “project manager server” is equivalent to the “controller,” and the job server is equivalent to the “task pool.”

Furthermore, as described above, “*job server is contacted by clients who pop elements from the queue and execute the request by downloading a script from the supplied server in the request*,” similarly, “*a first co-processor configured to successively: retrieve a first task from the task pool*,” (Patent '004, claim 1). Hence, the “job client” is equivalent to the “co-processor.”

The Folding@Home document cited above continues:

“Certificates



Four certificate sets are used in the **project management server** to **communicate with the job server and the client**:

- jobserver-cert.pem : Used as a CA to identify the job server. (To be replaced by a recognized CA)
- jobserver-pmca-cert.pem : A CA *certificate used by clients to identify the project management server.*"
(emphasis added, Reference 6, pages 1-2)

As stated above "project manager" (controller) communicates with the "client" (co-processor). Furthermore, the Folding@Home document cited above continues:

"Three certificates are utilized from within the client:

- jobserver-pmca-cert.pem : A CA *certificate which identifies and authorizes project management servers that the job client attempts to communicate with.*"
(emphasis added, Reference 6, page 2)

As stated above "job client" (co-processor) communicates with the "project manage server" (controller).

In conclusion, the BOINC architecture is fundamentally different from the architecture described by Patents '004 and '777, which requires no communication between the controller and the co-processor. The BOINC architecture, unequivocally requires a bilateral communication between the controller and the co-processor.

Furthermore, the SETI reference fails to disclose the claim requirement "on a plug-and-play basis" of the '004 patent. Nor does it disclose a "first agent ... being dispatched ... to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address... [and] returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address ..." as required by the '777 patent.

In view of the foregoing responses, Swarm still believes that a license under the Swarm patents should be of interest to Cisco. Accordingly, Swarm continues to be open to discussing license terms under conditions that would be favorable to an early licensee.

Sincerely yours,

John A. Fisher
IP licensing Consultant
Cc: Alfonso Iniguez

(480) 319-2233

phxfish@gmail.com

8300 S. Homestead Lane, Tempe, AZ 85284



Jennings, Strouss & Salmon, PLC
Attorneys at Law

One East Washington Street, Suite 1900
Phoenix, Arizona 85004-2554
Telephone: 602.262.5911
www.jsslaw.com

Michael K. Kelly
Direct Dial: 602.262.5824
Direct Fax: 602.495.2630
mkelly@jsslaw.com

October 3, 2018

Hilton Romanski
Senior Vice President and
Chief Strategy Officer
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

David Goeckeler
Executive Vice President and
General Manager
Networking and Security Business
Cisco System
170 West Tasman Drive
San Jose, CA 95134-1706

Re: Swarm Technology Licensing Opportunity

Dear Messrs. Romanski and Goeckeler:

The purpose of this letter is to follow up on our letter of August 8, 2018, in which we presented a licensing opportunity involving IoT and swarm intelligence technology developed by our client, Swarm Technology, LLC.

We continue to believe that Cisco is a particularly attractive candidate for this opportunity as Swarm's technology closely relates to Cisco's Open Plug-n-Play (PnP) Agent. As we understand it, Cisco's PnP Agent is an embedded software component present in certain Cisco network devices that support a simplified deployment architecture. The PnP agent interacts with a centralized PnP server to receive tasks, much in the same way that Swarm's intelligence agents interact with a central task pool. Your recent press releases confirm that this PnP Agent and Cisco IOS XE has been incorporated into more than one million ISR/ASR edge routers.

As detailed in our prior correspondence, Swarm's R&D efforts in IoT devices, edge computing, intent-based autonomy, plug-and-play robotics, and swarm processing systems have resulted in a number of valuable patent assets, such as U.S. Pat. No. 9,852,004, entitled "System and Method for Parallel Processing using Dynamically Configurable Proactive Co-Processing Cells," issued December 26, 2017 to Swarm; U.S. Patent No. 9,146,777, entitled "Parallel Processing With Solidarity Cells by Proactively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process," issued September 29, 2015 to Swarm; Pending U.S. Pat. App. Serial No. 15/852,480, entitled "System and Method for Swarm Intelligence using Dynamically Configurable Proactive Autonomous Agents," filed December 22, 2017; and various foreign

Phoenix ▶ Peoria ▶ Yuma ▶ Washington, DC


Hilton Romanski
David Goeckeler
October 3, 2018
Page 2

equivalents as detailed in our prior letter. We believe that Cisco could immediately leverage these assets to enhance Cisco's market share and product sales margins.

Swarm seeks to partner with an established company such as Cisco to accelerate industry adoption of Swarm's technology, and to monetize the aforementioned patent assets. We look forward to the opportunity to discuss exclusive or non-exclusive licensing arrangements with you.

Sincerely yours,

JENNINGS, STROUSS & SALMON, P.L.C.

By 
Michael K. Kelly

MKK/cm
cc: Alfonso Iniguez



Michael K. Kelly
Chair, Intellectual Property Department
P 602.262.5824 | F 602.495.2630
mkelly@jsslaw.com
Jennings, Strouss & Salmon, P.L.C.
One East Washington Street, Suite 1900
Phoenix, Arizona 85004-2554
jsslaw.com

August 7, 2018

Hilton Romanski
Senior Vice President and
Chief Strategy Officer
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

David Goeckeler
Executive Vice President and
General Manager
Networking and Security Business
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

Re: Swarm Technology, LLC

Dear Messrs. Romanski and Goeckeler:

The purpose of this letter is to highlight a licensing opportunity relating to IoT and swarm intelligence technology developed by our client, Swarm Technology, LLC. In short, Swarm's research and development efforts in IoT devices, edge computing, intent-based autonomy, plug-and-play robotics, and swarm processing systems (featured at www.swarmtechnology.us) have yielded a significant and growing global patent portfolio, which Swarm is currently seeking to license.

We believe your position in the enterprise software and hardware sectors makes you an attractive candidate for exploiting this opportunity. In particular, we think Cisco's Open Plug-n-Play (PnP) Agent as used in connection with your IOS devices would benefit from Swarm's technology, based on the documentation posted online for Cisco IOS XE Release 3E and later releases.

While non-exclusive licenses are preferred, Swarm will consider exclusive licensing arrangements within discrete product or industry segments. In an effort to jump start its licensing program, Swarm is willing to provide highly favorable terms to the first licensee.

Swarm's current patent assets include, for example:

1) U.S. Pat. No. 9,852,004, entitled "System and Method for Parallel Processing using Dynamically Configurable Proactive Co-Processing Cells," issued December 26, 2017 to Swarm;

2) U.S. Patent No. 9,146,777, entitled "Parallel Processing With Solidarity Cells by Proactively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process," issued September 29, 2015 to Swarm;

Hilton Romanski
David Goeckeler
August 7, 2018
Page 2

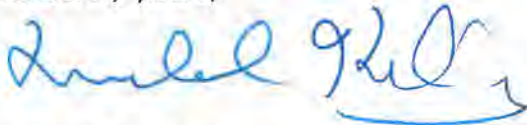
3) Pending U.S. Pat. App. Serial No. 15/852,480, entitled "System and Method for Swarm Intelligence using Dynamically Configurable Proactive Autonomous Agents," filed December 22, 2017; and

4) Various pending foreign applications, including European Pat. App. No. 15825.147.0-1879, Japanese Pat. App. No. 2017-503021, Chinese Pat. App. No. 201580039190, Indian Pat. App. No. 201747006602, and Hong Kong Pat. App. No. 17106684.9.

While Swarm's processing architecture is poised for rapid adoption by the IoT industry, Swarm remains a start-up company. Consequently, Swarm seeks to partner with an established company such as Cisco to quickly build market share and margins for products adopting Swarm's technology.

We would welcome the opportunity to discuss exclusive or non-exclusive patent licensing arrangements with you.

Sincerely yours,



Michael K. Kelly

MKK/mtl

cc: Alfonso Iniguez (via e-mail)

6226523v1(38145.4)

haynesboone

July 23, 2018

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

We are in receipt of your letter dated July 5, 2019 to Cisco Systems, Inc. ("Cisco").

Your letter references Swarm Technology LLC's current licensing efforts regarding a portfolio of patents. Your letter further includes a claim chart purporting to demonstrate the relevance of U.S. Patent No. 9,852,004 to Cisco's "IOS XE and the Cisco Plug and Play technology."

Cisco respects the intellectual property of others and takes such matters seriously. Our firm has been engaged to evaluate your concerns and any potential license offer. We are currently reviewing the chart provided with your letter, and we will follow up in due course.

Your letter also references previous letters directed to Hilton Romanski and David Goeckeler. Mr. Romanski is no longer with Cisco and Mr. Goeckeler's office has no record of receiving these letters. If you have further information about these letters, please provide it at your earliest convenience.

Please direct further communications to my attention.

Very truly yours,



Theodore Foster
972 739 8649 – Phone
theo.foster@haynesboone.com

4845-5710-5018 v.1

Haynes and Boone, LLP
Attorneys and Counselors
2323 Victory Avenue, Suite 700
Dallas, Texas 75219-7672
Phone: 214.651.5000
Fax: 214.651.5940

haynesboone

May 12, 2020

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

I write in response to your letter dated March 17, 2020 and your follow-up email dated May 4, 2020 discussing three patents, U.S. Patent No. 9,852,004 (the "'004 Patent"), U.S. Patent No. 9,146,777 (the "'777 Patent"), and U.S. Patent No. 10,592,275 (the "'275 Patent").¹ I address each patent in turn below.

'004 Patent

1. *Cisco's PnP devices are not "configured to . . . deliver the first task to the first co-processor."*

As noted in my previous letters, a Cisco Plug and Play ("PnP") device does not perform the "deliver" step as required of the claimed "first co-processor." Swarm's letter alleges that it is up to a "PnP Server to deliver a task."² But Swarm identifies the PnP Server as corresponding to the claimed "task pool," and not the claimed "first co-processor."³ Thus, Swarm's statement simply reaffirms that Swarm has not shown that a Cisco PnP device is "a first co-processor configured to . . . deliver the first task to the first co-processor."

¹ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Mar. 17, 2020) ("March 17, 2020 Letter").

² *Id.* at 3.

³ See John A. Fisher, *U.S. Patent 9,852,004 Claim Chart on U.S. 9,852,004*, at 1, attached to Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Jan. 8, 2020) ("'004 Claim Chart").

Haynes and Boone, LLP
Attorneys and Counselors
2505 North Plano Road, Suite 4000
Richardson, TX 75082
Phone: 214.651.5000

Letter to John A. Fisher
May 12, 2020

2. *Cisco's PnP devices do not join, and the Cisco PnP server does not "accept," Cisco PnP devices on a "plug-and-play basis."*

Claim 1 of the '004 Patent recites, in part, that "the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a **plug-and-play basis**."

In attempting to distinguish its claims from the prior art highlighted in Cisco's previous letters, Swarm explained that "manual installation, manual configuration and manual registration . . . defeat[s] the purpose of plug-and-play," and that such manual steps preclude a client from joining a server on a "plug-and-play" basis.⁴

Such manual installation, manual configuration and manual registration steps are required before a Cisco PnP device can be provisioned by the Cisco PnP server.

First, Swarm's previous letter indicates that a network administrator enters "pre-provisioning information" to the Cisco PnP server using a "deployment application."⁵ Swarm's letter alleges that the same *manual* configuration mechanism which is used to "pre-provision" Cisco PnP devices—the "deployment application"—is also used to "pre-provision" the Cisco PnP server:

The deployment application running on smart phones and PCs is used to pre-provision the PnP server since the same mechanism is alternatively used to manually configure the devices:

See '004 Claim Chart, at 2 (emphasis added). Thus, manual configuration is required to "pre-provision" the Cisco PnP server. Further, Swarm's previous letter alleges that such "pre-provisioning" is used to provide the Cisco PnP server with deployment related operational tasks, such as "[e]stablishing initial network connectivity for the device."⁶ Without manual configuration, it would not be possible for the Cisco PnP server to establish initial network connectivity with a Cisco PnP device and provide the Cisco PnP device with configuration information.

Second, the network administrator must manually download and install the "deployment application" on his or her smartphone or PC before any "pre-provisioning" information can be entered. Thus, manual installation is required before a Cisco PnP device can be provisioned by the Cisco PnP server.

Third, before a network administrator is permitted to download the "deployment application" from the Cisco website, the network administrator must register an account with Cisco and

⁴ See March 17, 2020 Letter, at 6.

⁵ '004 Claim Chart, at 2.

⁶ *Id.*

Letter to John A. Fisher
May 12, 2020

provide his or her log-in credentials to Cisco before any download will be authorized.⁷ Thus, manual registration is required before any “pre-provisioning information” can be entered to provision the Cisco PnP device.

Since manual installation, manual configuration and manual registration steps are required before a Cisco PnP device can be provisioned by the Cisco PnP server, Cisco’s PnP server is not “configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a **plug-and-play basis**,” under at least Swarm’s interpretation of “plug-and-play.”

Cisco previously presented additional distinctions between the claims and the identified Cisco PnP products. While Cisco continues to believe that those distinctions apply, reiteration is unnecessary in view of the analysis above.

’777 Patent

1. *Cisco’s PnP device does not have a “first agent.”*

Claim 1 of the ’777 Patent recites, in part, “wherein the first agent retrieves the matching task by: being dispatched . . . to the task pool . . . searching the task pool . . . and returning to the first solidarity cell.”⁸

Swarm states that the “claimed ‘agent’ **cannot be simply construed as a series of IPv4 datagrams.**”⁹ But Swarm’s allegations against Cisco’s PnP devices is dependent on just such an interpretation of the claim.¹⁰

Because Swarm cannot show the relevance of the ’777 Patent except under a claim interpretation that it disavows, it follows that Cisco’s PnP products do not include a “first agent [that] retrieves the matching task by: being dispatched . . . to the task pool . . . searching the task pool . . . and returning to the first solidarity cell.”¹¹

⁷ See, e.g., Cisco Software Download, [https://software.cisco.com/download/home/286208072/type/286291196/release/1.6%20\(PnP\)](https://software.cisco.com/download/home/286208072/type/286291196/release/1.6%20(PnP)) (last accessed Mar. 24, 2020).

⁸ ’777 Patent, claim 1 (in part).

⁹ March 17, 2020 Letter, at 7 (emphasis added).

¹⁰ See John A. Fisher, *U.S. Patent 9,146,777 Claim Chart*, at 2, attached to Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Jan. 8, 2020) (“’777 Claim Chart”) (alleging that the agent is implemented as an “interchange [] composed of two datagrams”).

¹¹ ’777 Patent, claim 1 (in part).

Letter to John A. Fisher
May 12, 2020

2. *Cisco's PnP device does not have a "first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task"*

Cisco also explained that claim 1 of the '777 Patent is distinguished by the claim requirement to retrieve a task "without requiring an instruction from the CPU."

As explained in Cisco's February 10, 2020 letter, Swarm's analysis in the '777 Claim Chart indicates that the Cisco FindIT Network Manager is run by a CPU (*i.e.*, the alleged "CPU"), and that the Cisco Plug and Play server (*i.e.*, the alleged "task pool") is "embedded" in the Cisco FindIT Network Manager.¹² Because the Cisco Plug and Play server is "embedded" in the Cisco FindIT Network Manager, the same CPU that runs the Cisco FindIT Network Manager also runs the Cisco Plug and Play server. Consequently, if the Cisco Plug and Play server performs a search for an eligible task, the CPU that runs the Cisco FindIT Network Manager would execute software instructions to perform that search.

Swarm's response does not dispute the facts of Cisco's position, but instead merely states "it is the **agent** that is to perform an action without instruction from the CPU."¹³ Swarm's response fails to address the distinguishing facts identified in Cisco's letter. Swarm does not show how the alleged agent is able to perform the action in question—retrieving a task—without requiring the Cisco PnP server executing on the alleged "CPU" to execute software instructions.

Cisco previously presented additional distinctions between the claims and the identified Cisco PnP products. While Cisco continues to believe that those distinctions apply, reiteration is unnecessary in view of the analysis above.

'275 Patent

1. *Cisco's PnP devices do not "operate in solidarity with each other"*

Claim 11 of the '275 Patent recites, in part, that the "plurality of autonomous co-processors operate in **solidarity** with each other and the task pool to complete the objective."

The '275 Patent specification explains that "[t]he term **solidarity** implies that co-processing cells share a common objective in **monitoring and executing all available tasks within the task pool**."¹⁴

¹² See Letter from Theo Foster, *Re: Swarm Technology Licensing Opportunity* (Feb. 10, 2020) ("February 10, 2020 Letter").

¹³ March 17, 2020 Letter, at 5.

¹⁴ '275 Patent, at 2:51-54 (emphasis added).

Letter to John A. Fisher
May 12, 2020

Additionally, the '275 Patent specification describes that “the solidarity cells 12A-12 n are **ambivalent** as to the particular composition of the thread itself.”¹⁵ This is because agents of the solidarity cells are “**only concerned about finding a match between the capabilities of its corresponding cell and an available task.**”¹⁶ The specification further elaborates that “**as long as** there are available tasks 22 in the task pool 13, and an available task 22 matches the capability of the cell, then the system may **effectively harness the processing capacity of the cell.**”¹⁷

In contrast, Cisco’s PnP devices are not concerned about matching their capabilities with tasks. Instead, as explained in Cisco’s previous letters, each of Cisco’s PnP devices “sends its **unique device identifier (UDI)** along with a request for work.”¹⁸ Thus, Cisco’s PnP devices are only concerned about obtaining and installing their *own* configurations from the Cisco PnP server.

Additionally, Cisco’s PnP devices do not execute available tasks “as long as” they are available. For example, even if a Cisco PnP device had the computing resources and processing capability to install another Cisco PnP device’s configuration, the Cisco PnP device would not do so as such a mistargeted installation would cause the Cisco PnP device to be incorrectly configured.

Thus, unlike the solidarity cells of the '275 Patent, Cisco’s PnP devices do not monitor or execute “all available tasks within the task pool.”¹⁹ Moreover, unlike the distributed processing system of the '275 Patent, neither Cisco’s intent-based networks nor Cisco’s PnP products “effectively harness the processing capacity of the cell” in the manner described in the '275 Patent.²⁰ Accordingly, Cisco’s PnP devices operate independently of one another and not “in **solidarity** with each other” as required by claim 11.

2. *Cisco’s PnP devices are not introduced on a “plug-and-play basis.”*

Claim 11 of the '275 Patent recites, in part, “dynamically introducing to an environment, on a **plug-and-play basis** . . . a plurality of autonomous co-processors.”

As explained above regarding the '004 Patent, manual installation, manual configuration and manual registration are required before a Cisco PnP device can be provisioned by the Cisco PnP server. However, Swarm regards such “manual installation, manual configuration and manual registration” as “defeat[ing] the purpose of plug-and-play” and precludes a client from joining the server on a “plug-and-play” basis.²¹

¹⁵ '275 Patent, at 9:22-23 (emphasis added).

¹⁶ *Id.* at 9:24-26 (emphasis added).

¹⁷ *Id.* at 9:26-29 (emphasis added).

¹⁸ Cisco, *Cisco Network Plug and Play Agent Configuration Guide, Cisco IOS XE Everest 16.6 12* (revised Oct. 30, 2017), <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/pnp/configuration/xe-16-6/pnp-xe-16-6-book.pdf>.

¹⁹ *See* '275 Patent, at 2:51-54.

²⁰ *See id.* at 9:26-29.

²¹ *See* March 17, 2020 Letter, at 6.

Letter to John A. Fisher
May 12, 2020

Thus, Cisco's PnP products do not "dynamically introduc[e] to an environment, on a **plug-and-play basis** . . . a plurality of autonomous co-processors" as required by claim 11.

For at least these reasons, Swarm's allegations regarding the '275 Patent are without merit.

Conclusion

In summary, Cisco has carefully considered and responded to your letter regarding the '004, '777, and '275 Patents. For the reasons established above and in the parties' prior communications, Cisco believes that a license would not be of benefit. Accordingly, Cisco is prepared to close this matter, and will do so absent further communication from you. Please direct any further communications to my attention.

Very truly yours,



Theo Foster
(972) 739-8649 – Phone
theo.foster@haynesboone.com

From: John Fisher <phxfish@gmail.com>
Sent: Thu, 22 Aug 2019 15:44:13 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: gayle.conner@haynesboone.com, Alfonso Íñiguez <alfonso@swarmtechnology.us>

Dear Mr. Foster:

It has been approximately one month since receiving your letter indicating that you would be evaluating the license offer and reviewing the claim chart sent to Cisco. I am writing now to inquire how that evaluation and review are progressing.

Mr. Iniquez and I are available and willing to discuss the license offer with Cisco at any convenient time.

John A. Fisher

From: John Fisher <phxfish@gmail.com>
Sent: Mon, 4 May 2020 12:59:53 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: "Conner, Gayle" <gayle.conner@haynesboone.com>, Alfonso Iñiguez <alfonso@swarmtechnology.us>
[Swarm Patent 275 Cisco Claim Chart Amended.pdf](#)

Dear Mr. Foster:

In my letter to you of March 17, 2020 I included a claim chart for Swarm's newly issued UP Patent 10,592,275. I have just discovered that there is an error in that claim chart. In printing the claim chart for transmission to you I accidentally truncated the claim chart, leaving off the end of the claim. I have attached a revised claim chart to this email. Please consider the revised claim chart and discard the claim chart sent previously.

I apologize if this has caused you any undue effort.

Respectfully,

John A. Fisher

haynesboone

August 27, 2019

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

I write to follow up on my July 23, 2019 letter that responded to your letter dated July 5, 2019 to Cisco Systems, Inc. ("Cisco").

As an initial matter, your letter states that you had earlier addressed letters dated August 8, 2018 and October 2, 2018 to Messrs. Romanski and Goeckeler at Cisco regarding a licensing opportunity. We appreciate you providing copies of those letters on July 24, 2019. However, Cisco does not have any records indicating that the letters were received by the abovementioned individuals or anyone else at Cisco on or around those dates. If you possess confirmation receipts of these letters, please kindly forward them to my attention.

Your July 5th letter also references Swarm Technology LLC's current licensing efforts regarding a portfolio of patents, and further includes a claim chart purporting to demonstrate the relevance of U.S. Patent No. 9,852,004 (the "'004 patent") to Cisco's "IOS XE and the Cisco Plug and Play technology."

Cisco respects the intellectual property of others and takes such matters seriously.

The claim chart accompanying your letter purports to map the claim language of the '004 patent to features found in Cisco's Network Plug and Play Agent Configuration Guide (the "Configuration Guide").¹ We have reviewed this claim chart carefully, and we note below

¹ Cisco, *Cisco Network Plug and Play Agent Configuration Guide, Cisco IOS XE Everest 16.6* (revised Oct. 30, 2017), available at <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/pnp/configuration/xe-16-6/pnp-xe-16-6-book.pdf>.

Haynes and Boone, LLP
Attorneys and Counselors
2505 North Plano Road, Suite 4000
Richardson, TX 75082
Phone: 214.651.5000

haynesboone

Letter to John A. Fisher
August 27, 2019

multiple errors and gaps in its analysis. To be clear, the deficiencies discussed below are merely exemplary, and additional issues exist.

First, your attempted mapping of the claimed “controller” of the ’004 patent to a human administrator is flawed, unsupported, and contrary to the disclosure of the patent. Nothing in the specification suggests that the “controller” could be a human, and it is doubtful that the Patent Office would have permitted the patent to issue had it known that Swarm Technology would take such an unreasonable view. As you are likely aware, courts have routinely found patent claims are invalid when they “seek to patent the use of human intelligence.” *In re Comiskey*, 499 F.3d 1365, 1379 (Fed. Cir. 2007).

As an example of this deficient mapping, claim 1 of the ’004 patent requires that the “controller” be “configured to populate the task pool.” The ’004 patent does not provide any description of how to “configure” a human to do this. Instead, the ’004 patent describes the “controller” as being a “CPU” or “a smartphone, tablet, laptop, or other device”—in other words, some kind of electronic device:

Referring now to FIG. 4, an internet of things network 400 includes a **controller (CPU) 402**, a task pool 408, and various devices 410-422, some or all of which include an associated or embedded microcontroller, such as an integrated circuit (IC) chip or other component which embodies processing capacity. By way of non-limiting example, the devices may include a light bulb 410, a thermostat 412, an electrical receptacle 414, a power switch 416, an appliance (e.g., toaster) 418, a vehicle 420, a keyboard 422, and virtually any other plug and play device or application capable of interfacing with a network.

In the illustrated embodiment, **the controller 402 may be a smartphone, tablet, laptop, or other device** which may include a display 404 and a user interface (e.g., keypad) 406 for facilitating user interaction with the various devices on the network. To the extent the processing capacity (e.g., bandwidth) of the controller 402 may be insufficient to adequately support the network, the controller may effectively harvest or recruit processing resources from the peripheral devices via the task pool, for example as explained below in conjunction with FIG. 5.²

The Federal courts recognize the differences between having a human perform a step and having a computer do it. In the words of one court, a device that operates on its own “works in a substantially different way” from a device that relies on a human. *Valley Recreation Prods., Inc. v. Arachnid, Inc.*, 35 U.S.P.Q.2d 1218, 1221 (Fed. Cir. 1994). That difference “precludes a finding of infringement.” *Id.* The district court in *Valley Recreation* found the patent owner’s allegations—which had ignored the differences between humans and computers—failed to

² ’004 patent, 11:35-55 (emphases added).



Letter to John A. Fisher
August 27, 2019

comply with Rule 11. Swarm's unsupported interpretation of the claimed "controller" warrants the same result.

Second, the Configuration Guide does not describe a "co-processor" as recited in claim 1 of the '004 patent. The patent specification explains how "co-processors" are "general purpose or special purpose processor[s]" that "work together **'in solidarity'** with one another and with the task pool to complete **aggregate** computational requirements by autonomously retrieving and completing individual tasks which may or may not be inter-related."³ The '004 patent goes on to explain that "[t]he term solidarity implies that co-processing cells **share a common objective** in monitoring and executing all available tasks within the task pool."⁴

The Configuration Guide explains that devices receiving the individual configurations from a Plug and Play server are not working "in solidarity" toward a "common objective," and thus, they are not "co-processors" as that term is used in the '004 patent claims. Unlike "co-processors," the Configuration Guide explains how each Cisco Plug and Play device requests its *own* configuration from the Cisco Plug and Play server using "a unique device identifier (UDI) along with a request for work."⁵ That is to say, the Configuration Guide describes a Cisco Plug and Play device requesting the configuration corresponding to its own unique and individual identifier. The Configuration Guide does not describe devices that are attempting to fulfill some "common objective." Thus, Cisco Plug and Play devices described in the Configuration Guide do not "work together 'in solidarity'" to complete "aggregate" computational requirements of a Cisco Plug and Play server. They are not described as engaging in any form of division of labor or "co-processing" with respect to one another or to the aggregate processing tasks of the Cisco Plug and Play server.⁶ Rather, the Configuration Guide describes how each Plug and Play device obtains its *individual* configuration information from the Cisco Plug and Play server. Accordingly, it is unreasonable to regard a Cisco Plug and Play device as a "co-processor" of claim 1 of the '004 patent.

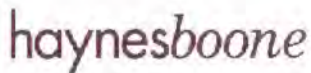
Third, the Configuration Guide does not describe a Cisco Plug and Play device as being "configured to... retrieve a first task from the task pool" as required by claim 1 of the '004 patent. The specification of the '004 patent describes that a co-processor performs such steps by dispatching an "agent" to the task pool to identify and gather a payload of tasks within the co-processor's capabilities:

³ '004 patent, 2:47-48 and 2:24-28 (emphases added). *See also* '004 patent, 6:39-42 (stating that the distributed processing system divides an "aggregate computational problem into a group of tasks").

⁴ '004 patent, 2:43-46 (emphasis added).

⁵ Configuration Guide at 12.

⁶ *See id.* at 5 (stating that "[w]hen a device is powered on for the first time, the PnP agent process wakes up in the absence of the startup config, user input on the device's console, and attempts to discover the address of the PnP server").



Letter to John A. Fisher
August 27, 2019

When an agent 30 is dispatched to the task pool 13 by its co-processor cell, the payload contains identifying information of the types of tasks the cell 12 can perform. When the agent 30 returns from the task pool 13, the payload contains the descriptor of the task 22, either in the form of a memory location or the full descriptor data structure.⁷

The '004 patent emphasizes that using agents to retrieve tasks will increase the processing capacity of the system because there is no "need to send a request ... to retrieve a task":

In other embodiments, some or all of the agents 30 are autonomous representatives of their respective corresponding cells 12. That is, **each agent 30 may be dispatched by its corresponding cell 12 to retrieve a task 22 any time the cell is idle or capable of performing additional processing.** In this way, the processing capacity of the solidarity cells 12 may be more fully exploited, inasmuch as the cells need not wait idly for an instruction from the CPU 11. This approach has the additional benefit of **reducing CPU overhead by relieving the CPU of the need to send a request to a cell to retrieve a task from the task pool.** These advantages render the system 10 more efficient than traditional computer architectures in which auxiliary modules and co-processors are dependent on instructions from the main CPU.⁸

In contrast, the Configuration Guide describes using the "send a request" approach that is specifically distinguished by the '004 patent. Specifically, the Configuration Guide explains that Cisco Plug and Play devices "sends... a request for work" to the Cisco Plug and Play server. Then, the Cisco Plug and Play server responds with "a work request" to be performed by the Cisco Plug and Play device:

The following steps indicate the Cisco Network Plug and Play agent deployment on Cisco devices:

- 1 The Cisco device, having PnP agent contacts the PnP server requesting for a task, that is, the **PnP Agent sends** its unique device identifier (UDI) along with a **request for work.**
- 2 **The PnP server if it has any task for the device, sends a work request. For example, image install, config upgrade, and so on.**
- 3 When the PnP agent receives the work request, executes the task and sends back a reply to the PnP server about the task status, whether it is a success or error, and the corresponding information requested.⁹

⁷ '004 patent, 8:53-58.

⁸ '004 patent, 8:59-9:5 (emphases added).

⁹ Configuration Guide at 12 (emphasis added).



Letter to John A. Fisher
August 27, 2019

While the Configuration Guide uses the word “agent,” it plainly refers to something very different from the “agents” described in the ’004 patent. The Cisco Network Plug and Play agent “is a software application that is running on a Cisco IOS or IOS-XE device”¹⁰ to be configured, and it remains on that device. In contrast with the agents described in the ’004 patent, the Configuration Guide does not describe the Cisco Network Plug and Play agent as ever leaving the device.

As such, the Cisco Network Plug and Play agent does not “retrieve” a task in the way described and claimed by the ’004 patent. Instead, the Cisco Network Plug and Play agent is described as operating like the traditional computer architectures distinguished in the ’004 patent in which a device must “wait idly for an instruction from the CPU.”¹¹ Specifically, the Configuration Guide describes how the Cisco Network Plug and Play agent sends a request and then must wait to receive an instruction from the Plug and Play server.

Accordingly, the Configuration Guide does not describe a “co-processor configured to ... retrieve a first task from a task pool” as recited by claim 1 of the ’004 patent.

Fourth, claim 1 requires a “first co-processor configured to... deliver the first task to the first co-processor.” Your claim chart fails to identify any evidence that a Cisco Plug and Play device (the alleged “first co-processor”) delivers a task to itself. Notably, the claim chart merely states that “The PnP server if it has any task for the device, sends a work request.”¹² It is our understanding, however, that the claim chart identified the PnP server as corresponding to the claimed “task pool,” *not* the claimed “first co-processor.” Thus, the claim chart fails to present evidence that a Cisco Plug and Play device could correspond to the “first co-processor configured to ... deliver the first task to the first co-processor,” as claimed.

For these exemplary reasons, we believe that the ’004 patent is not relevant to Cisco Plug and Play devices. We have reviewed the other patent mentioned in your letter (U.S. 9,146,777) and concluded that for similar reasons, it is not relevant to Cisco Plug and Play devices.

In view of the unreasonably broad interpretation that Swarm Technology has taken in its claim chart, we have also considered prior art that was not presented to or considered by the Patent Office before it decided to grant the ’004 patent. The prior art shows that, at a minimum, claim 1 of the ’004 patent is invalid as asserted. We will share our findings with you upon request.

We remain open to receiving from you an analysis identifying how you believe any of Cisco’s product offerings correspond to any claims of either patent, preferably in the form of a detailed

¹⁰ Configuration Guide at 5.

¹¹ ’004 patent, 8:65-66.

¹² Claim Chart at 2.

haynesboone

Letter to John A. Fisher
August 27, 2019

claim chart. It would be helpful, for example, if your analysis focuses on the features that were argued to the Patent Office as allegedly distinguishing the claims over the prior art.

Please direct further communications to my attention.

Very truly yours,



Theo Foster
(972) 739-8649 – Phone
theo.foster@haynesboone.com

From: John Fisher <phxfish@gmail.com>
Sent: Wed, 24 Jul 2019 11:27:03 -0700
Subject: Swarm Technology LLC's Patent Portfolio
To: theo.foster@haynesboone.com
Cc: gayle.connor@haynesboone.com, Alfonso Íñiguez <alfonso@swarmtechnology.us>
[#6311566v1 JSS-GEN - Swarm Ltr Romanski and Goeckeler 10-3-18.pdf](#)
[#6227880v1 JSS-GEN - Cisco Ltr to Romanski and Goeckeler 8-7-18.pdf](#)

Dear Mr. Foster:

Thank you for your email and letter of July 23, 2019.

Mr. Iniguez and I look forward to hearing from you after you have completed your review. In the meantime I have attached copies of the referenced letters to Romanski and Goeckeler.

John A. Fisher



Virus-free. www.avast.com



Jennings, Strouss & Salmon, PLC
Attorneys at Law

One East Washington Street, Suite 1900
Phoenix, Arizona 85004-2554
Telephone: 602.262.5911
www.jsslaw.com

Michael K. Kelly
Direct Dial: 602.262.5824
Direct Fax: 602.495.2630
mkelly@jsslaw.com

October 3, 2018

Hilton Romanski
Senior Vice President and
Chief Strategy Officer
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

David Goeckeler
Executive Vice President and
General Manager
Networking and Security Business
Cisco System
170 West Tasman Drive
San Jose, CA 95134-1706

Re: Swarm Technology Licensing Opportunity

Dear Messrs. Romanski and Goeckeler:

The purpose of this letter is to follow up on our letter of August 8, 2018, in which we presented a licensing opportunity involving IoT and swarm intelligence technology developed by our client, Swarm Technology, LLC.

We continue to believe that Cisco is a particularly attractive candidate for this opportunity as Swarm's technology closely relates to Cisco's Open Plug-n-Play (PnP) Agent. As we understand it, Cisco's PnP Agent is an embedded software component present in certain Cisco network devices that support a simplified deployment architecture. The PnP agent interacts with a centralized PnP server to receive tasks, much in the same way that Swarm's intelligence agents interact with a central task pool. Your recent press releases confirm that this PnP Agent and Cisco IOS XE has been incorporated into more than one million ISR/ASR edge routers.

As detailed in our prior correspondence, Swarm's R&D efforts in IoT devices, edge computing, intent-based autonomy, plug-and-play robotics, and swarm processing systems have resulted in a number of valuable patent assets, such as U.S. Pat. No. 9,852,004, entitled "System and Method for Parallel Processing using Dynamically Configurable Proactive Co-Processing Cells," issued December 26, 2017 to Swarm; U.S. Patent No. 9,146,777, entitled "Parallel Processing With Solidarity Cells by Proactively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process," issued September 29, 2015 to Swarm; Pending U.S. Pat. App. Serial No. 15/852,480, entitled "System and Method for Swarm Intelligence using Dynamically Configurable Proactive Autonomous Agents," filed December 22, 2017; and various foreign

Phoenix ▶ Peoria ▶ Yuma ▶ Washington, DC

Hilton Romanski
David Goeckeler
October 3, 2018
Page 2

equivalents as detailed in our prior letter. We believe that Cisco could immediately leverage these assets to enhance Cisco's market share and product sales margins.

Swarm seeks to partner with an established company such as Cisco to accelerate industry adoption of Swarm's technology, and to monetize the aforementioned patent assets. We look forward to the opportunity to discuss exclusive or non-exclusive licensing arrangements with you.

Sincerely yours,

JENNINGS, STROUSS & SALMON, P.L.C.

By 
Michael K. Kelly

MKK/cm
cc: Alfonso Iniguez



Michael K. Kelly
Chair, Intellectual Property Department
P 602.262.5824 | F 602.495.2630
mkelly@jsslaw.com
Jennings, Strouss & Salmon, P.L.C.
One East Washington Street, Suite 1900
Phoenix, Arizona 85004-2554
jsslaw.com

August 7, 2018

Hilton Romanski
Senior Vice President and
Chief Strategy Officer
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

David Goeckeler
Executive Vice President and
General Manager
Networking and Security Business
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706

Re: Swarm Technology, LLC

Dear Messrs. Romanski and Goeckeler:

The purpose of this letter is to highlight a licensing opportunity relating to IoT and swarm intelligence technology developed by our client, Swarm Technology, LLC. In short, Swarm's research and development efforts in IoT devices, edge computing, intent-based autonomy, plug-and-play robotics, and swarm processing systems (featured at www.swarmtechnology.us) have yielded a significant and growing global patent portfolio, which Swarm is currently seeking to license.

We believe your position in the enterprise software and hardware sectors makes you an attractive candidate for exploiting this opportunity. In particular, we think Cisco's Open Plug-n-Play (PnP) Agent as used in connection with your IOS devices would benefit from Swarm's technology, based on the documentation posted online for Cisco IOS XE Release 3E and later releases.

While non-exclusive licenses are preferred, Swarm will consider exclusive licensing arrangements within discrete product or industry segments. In an effort to jump start its licensing program, Swarm is willing to provide highly favorable terms to the first licensee.

Swarm's current patent assets include, for example:

1) U.S. Pat. No. 9,852,004, entitled "System and Method for Parallel Processing using Dynamically Configurable Proactive Co-Processing Cells," issued December 26, 2017 to Swarm;

2) U.S. Patent No. 9,146,777, entitled "Parallel Processing With Solidarity Cells by Proactively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process," issued September 29, 2015 to Swarm;

Hilton Romanski
David Goeckeler
August 7, 2018
Page 2

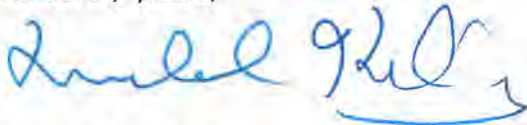
3) Pending U.S. Pat. App. Serial No. 15/852,480, entitled "System and Method for Swarm Intelligence using Dynamically Configurable Proactive Autonomous Agents," filed December 22, 2017; and

4) Various pending foreign applications, including European Pat. App. No. 15825.147.0-1879, Japanese Pat. App. No. 2017-503021, Chinese Pat. App. No. 201580039190, Indian Pat. App. No. 201747006602, and Hong Kong Pat. App. No. 17106684.9.

While Swarm's processing architecture is poised for rapid adoption by the IoT industry, Swarm remains a start-up company. Consequently, Swarm seeks to partner with an established company such as Cisco to quickly build market share and margins for products adopting Swarm's technology.

We would welcome the opportunity to discuss exclusive or non-exclusive patent licensing arrangements with you.

Sincerely yours,



Michael K. Kelly

MKK/mtl

cc: Alfonso Iniguez (via e-mail)

6226523v1(38145.4)

haynesboone

December 17, 2019

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

I write in response to your letter dated December 13, 2019 discussing two patents, U.S. Patent No. 9,852,004 (the "'004 patent") and U.S. Patent No. 9,146,777 (the "'777 patent").¹

In response to Cisco's explanations in its reply letters, it appears that Swarm has altered its interpretation or mapping of various claim elements from the analysis presented in your July 5, 2019 letter (regarding the '004 patent) and October 29, 2019 (regarding the '777 patent).

For example, in your July 5, 2019 letter, you mapped the claimed "controller" of the '004 patent to an "administrator" for the PnP server. But in your December 13, 2019 letter, you map the claimed "controller" to a "smartphone or PC."² As an example relating to the '777 patent, Swarm previously mapped the claimed "list of functions" to a product ID and a serial number.³ In your December 13, 2019 letter, however, you appear to be arguing that a product ID and serial number are equivalent to a list of functions.⁴

Considering the changes in Swarm's positions, we ask that Swarm provide an updated claim chart analysis explaining how you believe the '004 and '777 patent claims relate to any of Cisco's product offerings. It will be very helpful to our understanding of Swarm's current positions if the analysis includes an explanation of the relevance of any material quoted from Cisco's product documentation and, anywhere that Swarm is relying on the doctrine of

¹ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Dec. 13, 2019) ("December 13, 2019 Letter").

² *Id.* at 2.

³ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Oct. 29, 2019) at 18.

⁴ December 13, 2019 Letter at 8.

Haynes and Boone, LLP
Attorneys and Counselors
2505 North Plano Road, Suite 4000
Richardson, TX 75082
Phone: 214.651.5000

Letter to John A. Fisher
December 17, 2019

equivalents, an explanation of how an accused feature meets the function-way-result test or another approved legal framework for equivalency.⁵ We also ask that you include either copies of any references you cite or links to them online.

We look forward to receiving the requested information.

Very truly yours,



Theo Foster
(972) 739-8649 – Phone
theo.foster@haynesboone.com

⁵ See, e.g., *Warner-Jenkinson Co. v. Hilton Davis Chemical*, 520 U.S. 17 (1997).

haynesboone

February 10, 2020

Via FedEx and email to phxfish@gmail.com

John A. Fisher
IP Licensing Consultant
8300 S. Homestead Lane
Tempe, AZ 85284

Re: Swarm Technology LLC's Patent Portfolio

Dear Mr. Fisher:

I write in response to your letter dated January 8, 2020 and enclosed claim charts discussing two patents, U.S. Patent No. 9,852,004 (the "'004 patent") and U.S. Patent No. 9,146,777 (the "'777 patent").¹ I address each in turn below.

'004 Patent

Claim 1 of the '004 patent recites, in part, "a first co-processor configured to successively . . . deliver the first task to the first co-processor."²

But the analysis in the '004 claim chart³ states that the "task pool," and not the alleged "co-processor," performs the "deliver" step:

In recapitulation:

"a first co-processor configured to successively: retrieve [*this is the first datagram, which is originated by the co-processor*] a first task from the task pool; deliver [*this is the second datagram, which is originated by the task pool*] the first task to the first co-processor;" (emphasis and description added, Patent '004 Claim Elements)

'004 claim chart, at 4 (emphasis added).

¹ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* (Jan. 8, 2020) ("January 8, 2020 Letter").

² '004 patent, claim 1 (in part).

³ '004 patent claim chart enclosed with January 8, 2020 Letter ("'004 claim chart").

Haynes and Boone, LLP
Attorneys and Counselors
2505 North Plano Road, Suite 4000
Richardson, TX 75082
Phone: 214.651.5000

Letter to John A. Fisher
February 10, 2020

Swarm alleges that a Cisco Plug and Play device corresponds to the claimed “first co-processor,” and that a Cisco Plug and Play server corresponds to the claimed “task pool.” Thus, Swarm’s analysis alleges that a Cisco Plug and Play *server*—and not a Cisco Plug and Play *device*—performs the “deliver” step by originating a second datagram. Because a Cisco Plug and Play device does not perform the “deliver” step as required of the claimed “first co-processor,” Swarm has not shown that the ’004 patent is relevant to Cisco’s Plug and Play products.

Cisco previously presented additional distinctions between the claims and the identified Cisco Plug and Play products. While Cisco continues to believe that those distinctions apply, reiteration is unnecessary in view of the analysis above.

’777 Patent

Claim 1 of the ’777 patent recites, in part, “a first agent ... [that] retrieves the matching task by ... searching the task pool for a task.”⁴

Swarm alleges in its ’777 claim chart⁵ that the “first agent” is an IPv4 datagram sent from a Cisco Plug and Play device to a Cisco Plug and Play server.⁶ But an IPv4 datagram does not “search[] the task pool for a task” as required by claim 1. Swarm’s analysis does not dispute that fact, and indeed the ’777 claim chart merely indicates that the IPv4 datagram merely “*prompts the PnP Server to determine* which of the eligible task[s] to assign to the requesting device.”⁷ Since the analysis concedes that the PnP server itself performs any searching, Swarm has not identified an agent that “search[es] the task pool” as required by claim 1.

Furthermore, claim 1 also requires that the first agent search the task pool “without requiring an instruction from the CPU.”⁸ The ’777 claim chart identifies that the claimed “CPU” corresponds to the computer processor running Cisco FindIT Network Manager.⁹ That same computer processor also runs the software for the Cisco Plug and Play server, which is an embedded application within FindIT Network Manager.¹⁰ Thus, if the Cisco Plug and Play server performs a search for an eligible task, the computer processor would execute software instructions to perform that search. Thus, under Swarm’s analysis, the step of searching the task pool occurs *with* an instruction from the alleged “CPU”—not *without* such an instruction as recited in claim

⁴ ’777 patent, claim 1 (in part).

⁵ ’777 patent claim chart enclosed with Swarm’s January 8, 2020 Letter (“’777 claim chart”).

⁶ *Id.* at 2 (stating, “The IPv4 datagram – used by Cisco devices – maps to the concept of “agent” as defined in Patent ’777, which is the communication method used by the co-processor.”).

⁷ *Id.* at 5.

⁸ *See* ’777 patent, claim 1.

⁹ *See* ’777 patent claim chart, at 1 (citing Cisco, *Network Plug and Play Solution Guide for SMB 5* (last updated Jul. 29, 2019), https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/cisco-findIT-network-management/technical_reference/PnP_Guide_02.pdf (“SMB Solution Guide”)).

¹⁰ *See id.* at 1 (citing SMB Solution Guide, at 5).

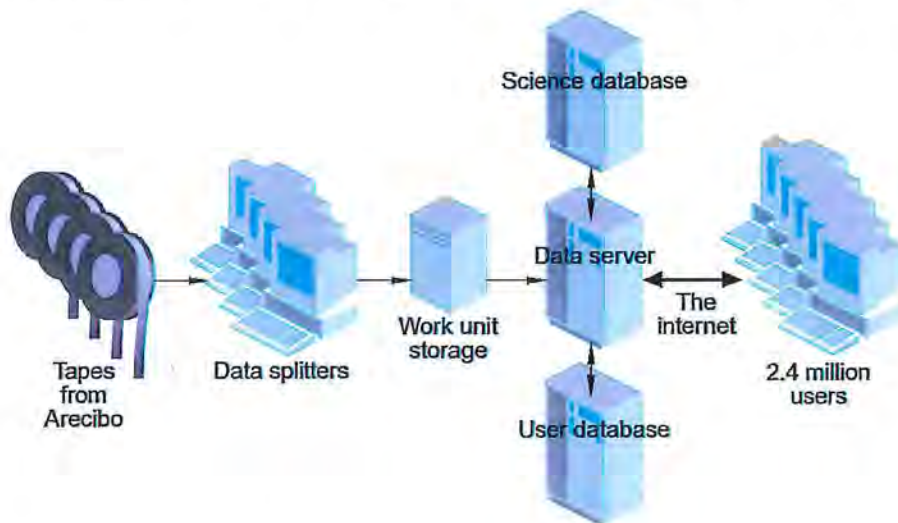
Letter to John A. Fisher
February 10, 2020

1. For this additional reason, Swarm has not shown that the '777 patent is relevant to Cisco's Plug and Play products.

Invalidity

Cisco previously provided a 2001 SETI@home article describing a processing architecture in which millions of computers collaborate to analyze astronomical data. Data splitters split the input astronomical data into small work units and place them into a work unit storage.¹¹ Each of the millions of users' computers have screen saver software that, when activated, communicate with the work unit storage to obtain and process work units. The users' computers do not communicate with the data splitters.¹²

Figure 2 of the 2001 SETI@home article (reproduced below) shows the architecture of the SETI@home system, including the abovementioned data splitters and work unit storage:



2001 SETI@home article, FIG. 2.

Swarm provided three responses to the 2001 SETI@home article.¹³ I address each in turn below.

¹¹ Letter from Theo Foster, *Re: Swarm Technology Licensing Opportunity* (Nov. 27, 2019) ("November 27, 2019 Letter") (citing Eric Korpela *et al.*, *SETI@Home—Massively Distributed Computing for SETI*, 3 *Computing in Science & Engineering* 79 (2001) ("2001 SETI@home article")).

¹² *Id.*

¹³ Letter from John A. Fisher, *Re: Swarm Technology Licensing Opportunity* 11 (Dec. 13, 2019) ("December 13, 2019 Letter") (citing David P. Anderson, *BOINC: A System for Public-Resource Computing and Storage*, in *Fifth IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, PA, 4-10 (Nov. 8, 2004)).

Letter to John A. Fisher
February 10, 2020

1. The original SETI@home project is not based on the BOINC architecture.

Much of Swarm's response addresses a different reference from that provided by Cisco. Swarm asserts that a later version of the SETI@home architecture was based on "BOINC" and was "fundamentally different" from the '004 and '777 patents.¹⁴ Any questions relating to the BOINC-based version of SETI@home are simply an irrelevant diversion, as the later version does nothing to diminish the relevance or applicability of the 2001 SETI@home article to Swarm's patent portfolio.

2. The 2001 SETI@home article describes that the SETI@home system accepts clients on a "plug-and-play" basis.

Claim 1 of the '004 patent recites, in part, a "processing system ... configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller."¹⁵

Swarm argues that the 2001 SETI@home article "fails to disclose the claim requirement 'on a plug-and-play basis' of the '004 patent."¹⁶

To the contrary, the 2001 SETI@home article describes how the SETI@home software "installs itself" by default as a screen saver. When the screen saver activates, the software joins the processing collective and begins to process data.¹⁷ Swarm has not identified any allegedly missing function required by the claims' "plug-and-play" limitation.

3. Communication between clients and the third server system is by HTTP.

Claim 1 of the '777 patent recites, in part:

the first agent ... being dispatched ... to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address ... [and] returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address....¹⁸

Swarm argues that the 2001 SETI@home article fails to disclose the above claim requirement.¹⁹

¹⁴ See December 13, 2019 Letter, at 9-11.

¹⁵ '004 patent, claim 1 (in part).

¹⁶ December 13, 2019 Letter, at 11.

¹⁷ 2001 SETI@home article, at 80.

¹⁸ '777 patent, claim 1 (in part).

¹⁹ December 13, 2019 Letter, at 11.

Letter to John A. Fisher
February 10, 2020

With that argument, Swarm appears to be taking two inconsistent positions regarding the meaning of the claimed “first agent” in the ’777 patent. In its accusations against Cisco’s products, Swarm alleges that the claimed “agent” is a series of IPv4 datagrams used for communication between devices. The 2001 SETI@home article describes how users’ computers communicate with the data server over the Internet using “**hypertext transfer protocol (HTTP)**.”²⁰ In 2001, it would have been obvious for HTTP messages to be carried over the Internet using IPv4 datagrams.²¹ Thus, to the extent that Swarm continues to allege that the “first agent” limitation covers using IPv4 datagrams for communication, Swarm cannot deny that the 2001 SETI@home article teaches or renders obvious the claimed “first agent.”

Conclusion

In summary, Cisco has carefully considered and responded to your updated claim charts regarding the ’004 and ’777 patents. For the reasons established above and in the parties’ prior communications, Cisco believes that a license would not be of benefit. Accordingly, Cisco is prepared to close this matter, and will do so absent further communication from you. Please direct any further communications to my attention.

Very truly yours,



Theo Foster
(972) 739-8649 – Phone
theo.foster@haynesboone.com

²⁰ 2001 SETI@home article, at 80 (emphasis added).

²¹ See, e.g., RFC 2616 at 13 (Jun. 1999) (“HTTP communication usually takes place over TCP/IP connections.”), <https://tools.ietf.org/html/rfc2616>.

From: John Fisher <phxfish@gmail.com>
Sent: Wed, 8 Jan 2020 16:01:06 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: Alfonso Íñiguez <alfonso@swarmtechnology.us>, "Conner, Gayle"
<gayle.conner@haynesboone.com>
[Cisco Tech Talks- FindIT PnP Overview and Configuration.pdf](#)
[IP Version 4 \(IPv4\) Datagram General Format.pdf](#)

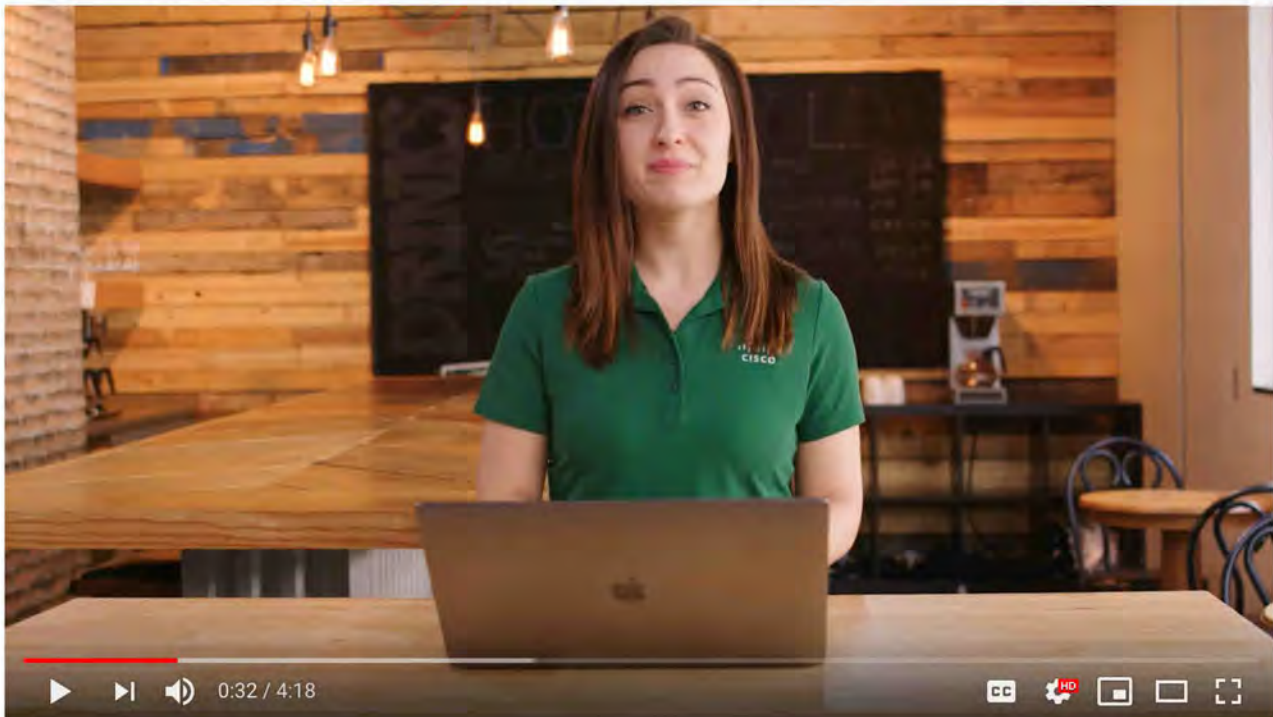
This is further to my email of earlier today. Mr Iniguez has suggested that it might avoid any confusion if I provided copies of the references that point to a web page or YouTube. Accordingly, I have attached copies of the Tech Talk and IPv4 references.

John Fisher

Cisco Tech Talks

FindIT PnP Overview & Configuration

Source: https://www.youtube.com/watch?v=DPE95_Kt_YM



Time Transcript

00:00 Cisco network Plug-and-Play or PnP
00:02 is a secure and scalable solution for
00:05 simple day zero provisioning across all Cisco
00:08 Enterprise platforms including routers
00:10 switches and wireless access points
00:13 we'll talk about the network PnP Server
00:15 and FindIT on this edition of Tech

00:18 Talks next with network PnP there are
00:25 four primary device management tools
00:27 we'll go over each of them in their
00:29 various uses and requirements start by
00:32 logging into your FindIT Network
00:34 Manager and navigating to the network
00:36 Plug-and-Play tab in the menu then go to
00:39 the images section to add an image into
00:42 the Manager you'll simply click the plus
00:44 icon in the upper left-hand corner of
00:46 the window here you can choose to either
00:49 drag and drop the image file or click in
00:53 the window to browse for a saved file
00:54 from your computer once we've done that
00:57 we can head to the configurations page
00:59 and click the plus icon in the corner to
01:02 upload our configuration file using the
01:04 same method as before
01:06 once our configuration file is uploaded
01:09 we can edit the product ID for our image
01:11 by going back to images selecting the
01:14 image then clicking the Edit icon in the
01:17 corner we can now move on to our first
01:19 management tool in the projects page
01:23 here we'll need to create our project by
01:26 entering a project name in the top left
01:28 corner and clicking create project once
01:31 the project is created
01:33 we'll need to provide a device name
01:35 Product ID and the serial number of the
01:37 device, I recommend finding the host name
01:40 for the device in the cisco network
01:42 probe and using that as your device name
01:45 the product ID and serial number can
01:48 also be found in the FindIT network
01:49 probe when you select a device and look
01:52 in the tab on the right once we've
01:55 entered all the appropriate information
01:56 we'll click add new to add it to the
01:59 projects device list at the bottom of
02:01 the page.
02:02 Now we'll select the check box next to
02:05 our recently added device and click the
02:08 Edit icon here we can add the image and
02:12 configuration
02:12 files we uploaded earlier from the image
02:15 and configure drop-down menus
02:17 then click the Save icon to save our
02:19 settings now we'll move on to the second
02:24 device management tool Auto claim to get
02:27 started we'll head to the auto claim

02:29 devices tab under Network Plug-and-Play. In
02:32 the menu, we'll enter our product ID into
02:35 the product ID box then click Add New,
02:38 now we'll select the device from the
02:41 list at the bottom of the page and click
02:43 the edit icon in the top left corner
02:45 once again we'll add our image and
02:48 configuration files from the image and
02:50 configure drop-down menus respectively
02:52 then save by clicking the Save icon in
02:55 the upper left corner, the network PnP
02:58 Server will use these files to provision
03:01 any device with a matching product ID
03:03 attempting to connect to the network. We
03:07 also have the option of manually
03:09 claiming or ignoring unclaimed devices
03:12 first navigate to the unclaimed devices
03:15 tab on the left-hand side to manually
03:18 claim a device we simply select it using
03:20 the check box from the list on the
03:22 unclaimed devices page we'll be sure to
03:25 assign our image and configuration files
03:28 to it then click on Play this will move
03:31 the device to the claimed list on the
03:34 unclaimed devices tab we can also choose
03:38 to ignore unclaimed devices by simply
03:40 checking the box next to the device and
03:42 clicking ignore we also have the
03:46 management option to adjust our check-in
03:48 time interval in the settings section
03:50 under the network Plug-and-Play tab this
03:53 interval determines how frequently our
03:55 PnP Server checks for updates just type
03:58 in your desired interval in minutes and
04:00 click Save to save your settings now
04:04 you're ready to configure your network
04:05 Plug-and-Play Server thanks for watching
04:08 Tech Talks from Cisco we'll see you next
04:10 time.

The TCP/IP Guide

A TCP/IP Reference You Can Understand!

IP Datagram General Format

(Page 1 of 3)

Data transmitted over an internet using IP is carried in messages called *IP datagrams*. Like all network protocol messages, IP uses a specific format for its datagrams. We are of course looking here at [IP version 4](#) and so we will examine the IPv4 datagram format, which was defined in RFC 791 along with the rest of IPv4.

The IPv4 datagram is conceptually divided into two pieces: the *header* and the *payload*. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. Unlike some message formats, IP datagrams do not have a footer following the payload.

Even though IP is a relatively simple, connectionless, "unreliable" protocol, the IPv4 header carries a fair bit of information, which makes it rather large. At a minimum, it is 20 bytes long, and with options can be significantly longer. The IP datagram format is described in [Table 56](#) and illustrated in [Figure 86](#).

Table 56: Internet Protocol Version 4 (IPv4) Datagram Format

Field Name	Size (bytes)	Description												
Version	1/2 (4 bits)	Version: Identifies the version of IP used to generate the datagram. For IPv4, this is of course the number 4. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP. In general, a device running an older version of IP will reject datagrams created by newer implementations, under the assumption that the older version may not be able to interpret the newer datagram correctly.												
IHL	1/2 (4 bits)	Internet Header Length (IHL): Specifies the length of the IP header, in 32-bit words. This includes the length of any options fields and padding. The normal value of this field when no options are used is 5 (5 32-bit words = 5*4 = 20 bytes). Contrast to the longer <i>Total Length</i> field below.												
TOS	1	Type Of Service (TOS): A field designed to carry information to provide quality of service features, such as prioritized delivery, for IP datagrams. It was never widely used as originally defined, and its meaning has been subsequently redefined for use by a technique called <i>Differentiated Services (DS)</i> . See below for more information.												
TL	2	Total Length (TL): Specifies the total length of the IP datagram, in bytes. Since this field is 16 bits wide, the maximum length of an IP datagram is 65,535 bytes, though most are much smaller.												
Identification	2	Identification: This field contains a 16-bit value that is common to each of the fragments belonging to a particular message; for datagrams originally sent unfragmented it is still filled in, so it can be used if the datagram must be fragmented by a router during delivery. This field is used by the recipient to reassemble messages without accidentally mixing fragments from different messages. This is needed because fragments may arrive from multiple messages mixed together, since IP datagrams can be received out of order from any device. See the discussion of IP message fragmentation.												
Flags	3/8 (3 bits)	<p>Flags: Three control flags, two of which are used to manage fragmentation (as described in the topic on fragmentation), and one that is reserved:</p> <table border="1"> <thead> <tr> <th>Subfield Name</th><th>Size (bytes)</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Reserved</td><td>1/8 (1 bit)</td><td>Reserved: Not used.</td></tr> <tr> <td>DF</td><td>1/8 (1 bit)</td><td>Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.</td></tr> <tr> <td>MF</td><td>1/8 (1 bit)</td><td>More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.</td></tr> </tbody> </table>	Subfield Name	Size (bytes)	Description	Reserved	1/8 (1 bit)	Reserved: Not used.	DF	1/8 (1 bit)	Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.	MF	1/8 (1 bit)	More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.
Subfield Name	Size (bytes)	Description												
Reserved	1/8 (1 bit)	Reserved: Not used.												
DF	1/8 (1 bit)	Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.												
MF	1/8 (1 bit)	More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.												
Fragment Offset	1 5/8 (13 bits)	Fragment Offset: When fragmentation of a message occurs, this field specifies the offset, or position, in the overall message where the data in this fragment goes. It is specified in units of 8 bytes (64 bits). The first fragment has an offset of 0. Again, see the discussion of fragmentation for a description of how the field is used.												
TTL	1	<p>Time To Live (TTL): Short version: Specifies how long the datagram is allowed to "live" on the network, in terms of router hops. Each router decrements the value of the TTL field (reduces it by one) prior to transmitting it. If the TTL field drops to zero, the datagram is assumed to have taken too long a route and is discarded.</p> <p>See below for the longer explanation of <i>TTL</i>.</p>												

Protocol	1	<p>Protocol: Identifies the higher-layer protocol (generally either a transport layer protocol or encapsulated network layer protocol) carried in the datagram. The values of this field were originally defined by the IETF "Assigned Numbers" standard, RFC 1700, and are now maintained by the Internet Assigned Numbers Authority (IANA):</p> <table border="1"> <thead> <tr> <th>Value (Hexadecimal)</th><th>Value (Decimal)</th><th>Protocol</th></tr> </thead> <tbody> <tr><td>00</td><td>0</td><td>Reserved</td></tr> <tr><td>01</td><td>1</td><td>ICMP</td></tr> <tr><td>02</td><td>2</td><td>IGMP</td></tr> <tr><td>03</td><td>3</td><td>GGP</td></tr> <tr><td>04</td><td>4</td><td>IP-in-IP Encapsulation</td></tr> <tr><td>06</td><td>6</td><td>TCP</td></tr> <tr><td>08</td><td>8</td><td>EGP</td></tr> <tr><td>11</td><td>17</td><td>UDP</td></tr> <tr><td>32</td><td>50</td><td>Encapsulating Security Payload (ESP) Extension Header</td></tr> <tr><td>33</td><td>51</td><td>Authentication Header (AH) Extension Header</td></tr> </tbody> </table> <p>Note that the last two entries are used when IPsec inserts additional headers into the datagram: the AH or ESP headers.</p>	Value (Hexadecimal)	Value (Decimal)	Protocol	00	0	Reserved	01	1	ICMP	02	2	IGMP	03	3	GGP	04	4	IP-in-IP Encapsulation	06	6	TCP	08	8	EGP	11	17	UDP	32	50	Encapsulating Security Payload (ESP) Extension Header	33	51	Authentication Header (AH) Extension Header
Value (Hexadecimal)	Value (Decimal)	Protocol																																	
00	0	Reserved																																	
01	1	ICMP																																	
02	2	IGMP																																	
03	3	GGP																																	
04	4	IP-in-IP Encapsulation																																	
06	6	TCP																																	
08	8	EGP																																	
11	17	UDP																																	
32	50	Encapsulating Security Payload (ESP) Extension Header																																	
33	51	Authentication Header (AH) Extension Header																																	
Header Checksum	2	<p>Header Checksum: A checksum computed over the header to provide basic protection against corruption in transmission. This is not the more complex CRC code typically used by data link layer technologies such as Ethernet; it's just a 16-bit checksum. It is calculated by dividing the header bytes into words (a word is two bytes) and then adding them together. The data is not checksummed, only the header. At each hop the device receiving the datagram does the same checksum calculation and on a mismatch, discards the datagram as damaged.</p>																																	
Source Address	4	<p>Source Address: The 32-bit IP address of the originator of the datagram. Note that even though intermediate devices such as routers may handle the datagram, they do not normally put their address into this field—it is always the device that originally sent the datagram.</p>																																	
Destination Address	4	<p>Destination Address: The 32-bit IP address of the intended recipient of the datagram. Again, even though devices such as routers may be the intermediate targets of the datagram, this field is always for the ultimate destination.</p>																																	
Options	Variable	<p>Options: One or more of several types of options may be included after the standard headers in certain IP datagrams. I discuss them in the topic that follows this one.</p>																																	
Padding	Variable	<p>Padding: If one or more options are included, and the number of bits used for them is not a multiple of 32, enough zero bits are added to "pad out" the header to a multiple of 32 bits (4 bytes).</p>																																	
Data	Variable	<p>Data: The data to be transmitted in the datagram, either an entire higher-layer message or a fragment of one.</p>																																	

Source: http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm

From: John Fisher <phxfish@gmail.com>
Sent: Tue, 3 Sep 2019 12:20:46 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: "Conner, Gayle" <gayle.conner@haynesboone.com>, Alfonso Íñiguez <alfonso@swarmtechnology.us>
[Cisco response 09032019 pdf.pdf](#)

I have attached a response to your letter of August 27, 2019.
John A. Fisher



September 3, 2019

Via US Mail and email to theo.foster@haynesboone.com

Theo Foster
Haynes and Boone, LLP
2505 North Plano Road, Suite 4000
Richardson, TX 75082

Re: Swarm Technology Licensing Opportunity
FRE 408

Dear Mr. Foster:

Your letter of August 27, 2019 voices four objections to the Swarm claim chart sent to Cisco Systems. This letter will address each of those objections.

First:

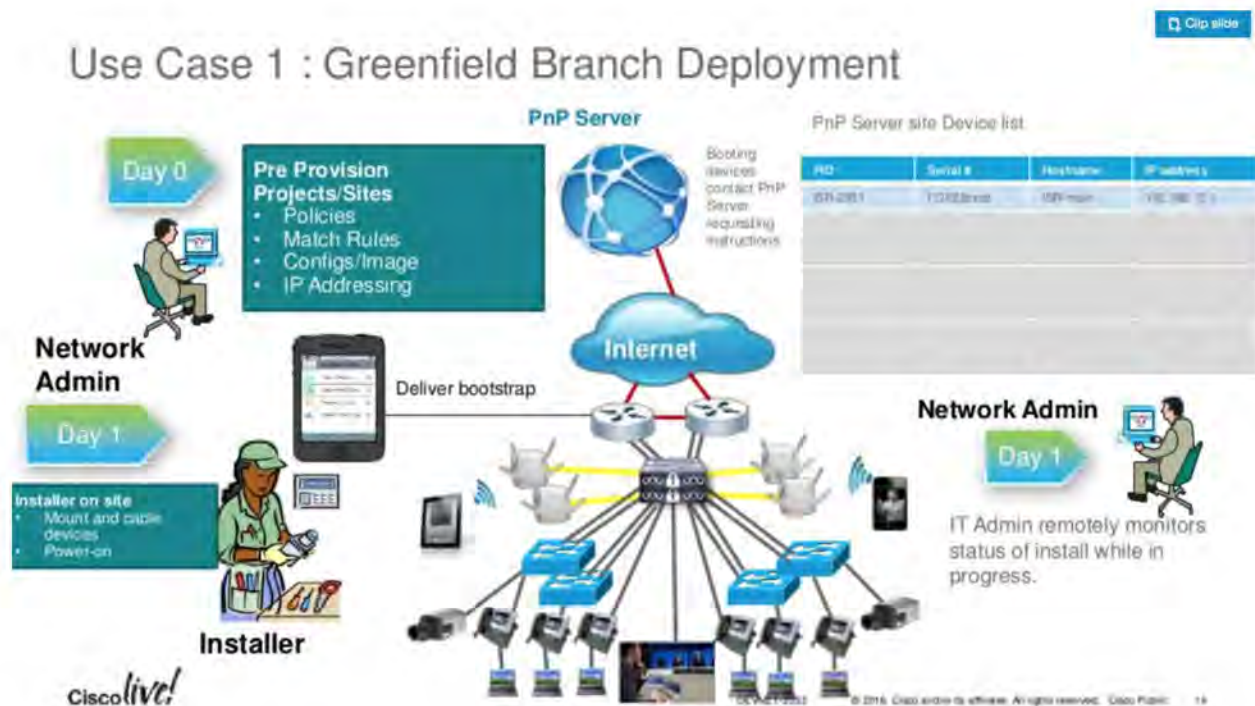
You stated that the claim chart "attempted mapping of the claimed 'controller' of the '004 patent to a human...."

Compare the Swarm device to the Cisco device as disclosed in the Cisco literature:

In the illustrated embodiment, **the controller 402 may be a smartphone, tablet, laptop, or other device which may include a display 404 and a user interface (e.g., keypad) 406 for facilitating user interaction with the various devices on the network.** To the extent the processing capacity (e.g., bandwidth) of the controller 402 may be insufficient to adequately support the network, the controller may effectively harvest or recruit processing resources from the peripheral devices via the task pool, for example as explained below in conjunction with FIG. 5. (emphasis added, '004 Patent, Column 11, lines 46-55)



It is true that the claim chart pointed to an "Administrator" in the Cisco reference. Cisco references, however, indicate that the Administrator is not merely a human, but rather is exactly what the '004 patent discloses. The following figure, from Cisco's documentation, shows the Network Administrator with a user interface entering the policies and configuration into the PnP server.



"Device Programmability with Cisco Plug-n-Play Solution" (slide #14)

Source: <https://www.slideshare.net/CiscoDevNet/device-programmability-with-cisco-plugnplay-solution>

In a similar manner Figure 2 of the Cisco Network Plug and Play Agent Configuration Guide shows the Network Administrator as a smartphone or laptop **for facilitating user interaction with the various devices on the network.** In



each example the human is interfacing with a computer that includes a display and a user interface exactly as disclosed in the '004 patent.

Second:

You state that the Cisco devices do not work to fulfill some common objective and do not work together to complete aggregate computational requirements and thus are not co-processors as claimed. You support this by quoting: "the Configuration Guide explains how each Cisco Plug and Play device requests its *own* configuration from the Cisco Plug and Play server using a 'unique device identifier (UDI) along with a request for work.'" Similar language exists in the '004 patent to define its agents: "plurality of devices each having a **unique dedicated agent** configured to proactively retrieve a task from the pool without direct communication with the CPU." (emphasis added, '004 patent, column 13, lines 62-64)

Although you state that "The Configuration Guide does not describe devices that are attempting to fulfill some 'common objective'", that point is refuted by Cisco's own words:

"The main principle is that the network takes the business intent and automatically transforms it into network configurations for all the devices." (Benefits of Intent-Based Networking, page 2)

"[A]ll the network devices will be automatically configured to **fulfill** this requirement across the network." (emphasis added, Benefits of Intent-Based Networking, page 2-3)



Third:

You state that the Cisco Plug and Play device is not configured to retrieve a first task from the task pool as required by claim 1. Specifically you state:

"While the Configuration Guide uses the word 'agent,' it plainly refers to something very different from the 'agents' described in the '004 patent. The Cisco Network Plug and Play agent 'is a software application that is running on a Cisco IOS or IOS-XE device' to be configured, and it remains on that device. In contrast with the agents described in the '004 patent, the Configuration Guide does not describe the Cisco Network Plug and Play agent as ever leaving the device."

I agree that Swarm and Cisco use the word "agent" to refer to different entities. I disagree, however, with your characterization in the foregoing paragraph that the Cisco agent remains on the device and the '004 agent leaves the device.

At issue is how the co-processor (in '004 language) or the plug and play agent (in Cisco language) communicates with the task pool ('004 language) or PnP Server (Cisco language), and specifically how the PnP agent retrieves a task from the PnP Server.

As clearly stated, the Configuration Guide defines the term "Plug and Play agent" as a software application that runs on the device. Furthermore, the Configuration Guide explains that "The PnP agent uses methods like DHCP, Domain Name System (DNS), and others to acquire the desired IP address of the PnP server".
(page 5)

Even though the Configuration Guide does not use the word "agent" to describe how the Plug and Play agent communicates with the PnP server, Cisco does say that the PnP agent relies on DHCP packets, and others, as methods of communication. This is explained here: "A DHCP relay agent is any host that forwards DHCP packets between clients and servers."



(IP Addressing: DHCP Configuration Guide, Cisco IOS Release 15SY Source: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dhcp/configuration/15-sy/dhcp-15-sy-book/dhcp-overview.html#GUID-D119F759-65F2-437F-B569-D9049145DA35)

Therefore Cisco's DHCP packets are analogous to the agent defined in the '004 patent, i.e., "the term *agent* refers to a software module, analogous to a *network packet*, associated with a co-processor that interacts with the task pool" (emphasis added, '004 patent, column 3, lines 13-14).

Consequently, by means of using network packets, the Plug and Play agent is able **contact**, **request**, **send**, and **receive** information from the PnP server as required by claim 1.

"1 The Cisco device, having PnP agent **contacts** the PnP server **requesting** for a task, that is, the PnP Agent **sends** its unique device identifier (UDI) along with a request for work.

2 The PnP server if it has any task for the device, **sends** a work request. For example, image install, config upgrade, and so on.

3 When the PnP agent **receives** the work request, executes the task and **sends** back a reply to the PnP server about the task status, whether it is a success or error, and the corresponding information requested." (emphasis added page 11)

In each instance the above actions of contact, request, send, send, receive, and send are implemented by an agent of the Cisco device, namely the DHCP packets, or other network packet communication methods.



Forth:

You say that the claim chart fails to present evidence of a Cisco PnP device corresponding to a first co-processor configured to deliver a first task to the first co-processor.

The claim language "a first co-processor configured to successively: ... deliver the first task to the first co-processor..." is explained, for example, in one exemplary embodiment, in column 13 in the sentence beginning at line 13. "The method includes the steps of: ... proactively sending a first agent from the first cell to the task pool; searching the task pool, by the first agent, for a task of the first type; retrieving, by the first agent, the first task from the task pool; transporting, by the first agent, the first task to the first cell...." Further, "the term agent refers to a software module, analogous to a network packet, associated with a co-processor that interacts with the task pool to thereby obtain available tasks which are appropriate for that co-processor cell." (Col 4, line 13) As explained above, the '004 agent associated with a particular co-processor is analogous to the DHCP packet associated with a particular Cisco PnP agent.

In the Cisco system the Cisco device automates its own configuration as set forth in the section "Information About Cisco Network Plug and Play Agent" (Page 5). "The Cisco Network PnP Agent is a part of Cisco Network Plug and Play solution.... Simplified deployment process of any Cisco device automates the following deployment related operational tasks:

....

•Delivering device configuration."

That is, the Cisco device, through the action of its agent, e.g., a DHCP packet, automatically delivers a task such as device configuration to the co-processor.

Accordingly, Swarm believes that the issues you have raised are without merit. You stated that other issues exist beyond the four you specifically enumerated.



Swarm would welcome the opportunity to discuss any such additional issues with you. You have also said that you have prior art that may be relevant to the '004 patent that you are willing to share with us. We, of course, would like to review any art that you have found.

Swarm continues to believe that the '004 patent is relevant to product offered by Cisco and we remain willing to discuss a licensing opportunity with Cisco. As mentioned in past correspondence, most favorable licensing terms will be available to early licensees.

Sincerely yours,

John A. Fisher
IP licensing Consultant

Cc: Alfonso Iñiguez

From: John Fisher <phxfish@gmail.com>
Sent: Wed, 8 Jan 2020 13:42:05 -0700
Subject: Swarm Technology Licensing Opportunity
To: theo.foster@haynesboone.com
Cc: "Conner, Gayle" <gayle.conner@haynesboone.com>, Alfonso Íñiguez <alfonso@swarmtechnology.us>
[Swarm Patent 777 Cisco Claim Chart 010820.pdf](#)
[Swarm Patent 004 Cisco Claim Chart Updated.pdf](#)
[cisco 010820.docx](#)
[citations.docx](#)

I have attached a letter to Mr. Theo Foster, two claim charts, and a list of citations.
John Fisher



January 8, 2020

Via US Mail and email to theo.foster@haynesboone.com

Theo Foster
Haynes and Boone, LLP
2505 North Plano Road, Suite 4000
Richardson, TX 75082

Re: Swarm Technology Licensing Opportunity FRE 408

Dear Mr. Foster:

This is in response to your letter of December 17, 2019. At the end of this letter I have attached revised claim charts for US Patents 9,852,004 and 9,146,777. These revised claim charts provide detailed analysis of the correspondence between Cisco products, as described in various pieces of Cisco literature, and the claims of the Swarm patents.

I have also attached a list of references relied upon and links to them online. I apologize for not providing these links before, but I had assumed, since all but one of the references are Cisco documents, that the references would be readily available to you.

Swarm continues to believe that the '004 and '777 patent are relevant to Cisco products. As I have said in past letters, Swarm is willing to discuss a license agreement under the Swarm patents with terms that will be most favorable to early licensees.

Sincerely yours,

John A. Fisher
IP licensing Consultant
Cc: Alfonso Iniguez

(480) 319-2233

phxfish@gmail.com

8300 S. Homestead Lane, Tempe, AZ 85284

Cisco Network Plug and Play Agent Configuration Guide, Cisco IOS XE Everest 16.5.1b

<https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/pnp/configuration/xe-16-6/pnp-xe-16-6-book.pdf>

Cisco FindIT Network Manager and Probe Administration Guide, Version 2.0

https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/cisco-findIT-network-management/admin_guide/b_Cisco_FindIT_Network_Management_Admin_2_0.pdf

Network Plug and Play Solution Guide for SMB

https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/cisco-findIT-network-management/technical_reference/PnP_Guide_02.pdf

IP Version 4 (IPv4) Datagram General Format

http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm

Cisco Understanding Virtual Network Function Descriptors

https://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/4-4/user/guide/Cisco-Elastic-Services-Controller-User-Guide-4-4/understanding_virtual_network_function_descriptors.pdf

Cisco Managing Configuration Files

<https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/config-mgmt/configuration/xe-3se/cat3850/config-mgmt-xe-3se-cat3850-book/cm-config-files.pdf>

Cisco Tech Talks: FindIT PnP Overview and Configuration

https://www.youtube.com/watch?v=DPE95_Kt_YM